

PAGE 6/85 * RCVD AT 10/2/2007 10:50:59 PM [Eastern Daylight Time] * SVR:USPTO-EFXRF-5/21 * DNIS:2738300 * CSID:714 540 9823 * DURATION (mm:ss):12:34

request to the structure generating means 120 in order to generate a desired structure body from in a structure body group for constituting each specific structure, from a user. The structure generating means 120 generates and presents a specific logical structure so as to satisfy the constraint condition in accordance with the instruction request from the user inputted through the generation instructing part 140, and also, stores it in a specific structure store part 132.

LEGAL STATUS [Date of request for examination] 21.08.1997
[Date of sending the examiner's decision of rejection] 06.11.2001
[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]
[Date of final disposal for application]
[Patent number]
[Date of registration]
[Number of appeal against examiner's decision of rejection]
[Date of requesting appeal against examiner's decision of rejection]
[Date of extinction of right]

* NOTICES *

JPO and INPIT are not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.

2. **** shows the word which can not be translated.

3. In the drawings, any words are not translated.

CLAIMS

[Claim(s)]

[Claim 1] Based on the constraint imposed on logical structure information common to two or more documents, the logical structure information on a specific document is interactively set to the document processing system equipment in which generation and edit are possible. While generating the candidate of the logical structure information on all generable specific documents who satisfies the constraint which shows said constraint and showing with the constraint concerned Document processing system equipment characterized by having a structure generation means to generate and show one logical structure information according to a demand out of the shown candidate.

DETAILED DESCRIPTION

[Detailed Description of the Invention]

[0001]

[Industrial Application] This invention relates to the document processing system equipment which has the function to generate and edit interactively the specific-logical structure which is the logical structure information on a specific document, filling the constraint on the structure by the common logical structure which is logical structure information common to two or more documents.

[0002]

[Description of the Prior Art] In recent years, an electronic document spreads and the technique of automatic processing of documents, such as automatic generation of documents, such as automatic allocation of a document and an extract from two or more

documents, is becoming important. However, the document set as the object of automatic processing in order to perform automatic processing of the document of a large quantity needs to have the same description. In order to attain such an object, the document expressed by common structure with information common to two or more documents and specific structure with the information on a specific document is required. As such a document, there is a document in ODA (opening document architecture: ISO 8613=Open Document Architecture) which is an international standard. This document consists of four structures of common logical structure and common allocation structure, specific-logical structure, and specific allocation structure, and the logical structure consists of combination of the component part of documents, such as a chapter, a knot, and a paragraph.

[0003] Moreover, in the conventional structured statement document editor, the system with which evaluates the constraint at the event to the creation time of document structure, and it is provided automatically [the candidate of a generable component part / one] is proposed. Sequential assessment of the subordination child production rule which can be adapted in process of document generation is carried out, consequently he shows a generable component part, and is trying to look for decision to a user in such a system.

[0004] For example, in "Grif" (glyph) ("Interactively editing structured documents" Richard FURUTA, Vincent QUINT, and Jacques ANDRE Electronic Publishing Vol.1(1)) which is one of the structuring editors, the Boolean part article which can be added and inserted is restricted before and after that by the logical structure of the document which is observing at the time of edit.

[0005] For example, as shown in drawing 36, it sets in Chapter 2 which consists of four knots, "Section 2.1", "Section 2.2", "Section 2.3", and "Section 2.4." If additional insertion of components is operated paying attention to Section 2.4 (Section 4 of Chapter 2), as a candidate in whom additional insertion is possible, low order knot components will be shown chapter components] in the knot components ahead of [, and back, and knot components and back will be shown it back. These candidates show that the additional insertion of "2+" can be carried out as "2.4+" and back chapter components as "2.4.1" and back knot components as "2.4 -" and back low order knot components as front knot components, respectively in drawing 36.

[0006] Moreover, when an observing point is Section 2.3, as a candidate in whom additional insertion is possible, the knot components ahead of [and back are shown low order knot components, and knot components] is shown back. These candidates show that "2.3.1" is carried out in "2.3 -" and back, and they can carry out additional insertion of "2.3+" as knot components back as low order knot components as front knot components in drawing 36, respectively. In addition, since Section 2.4 exists back when "Section 2.3" is observed, division of the chapter (Chapter 2) by insertion of chapter components has been forbidden.

[0007] The example of a display of the candidate in whom such additional insertion is possible is shown in drawing 37 and drawing 38.

[0008] In the example shown in drawing 37, are displayed on the window 3700 displayed on the display screen. In Chapter 2 (2Second Section) which consists of four knots, "Section 2.1 CCCCC", "Section 2.2 DDDDD", "Section 2.3 EEEEE", and "Section

2.4 FFFFFF" As a result of operating additional insertion of the components to "Section 2.4", the list of the candidate in whom additional insertion is possible is displayed on the window 3710. the inside of the candidate currently displayed on the window 3710 -- "Within selection" -- "Section level 2 before" means [low order knot components] back that knot components are carried out back and, as for "Section level 1 after", "Section level 2 after" can carry out additional insertion of the **** components back, respectively for knot components ahead.

[0009] Also in the example shown in drawing 38 , in Chapter 2 shown in drawing 37 , as a result of operating additional insertion of the components to "Section 2.3", the list of the candidate in whom additional insertion is possible is displayed on the window 3810.

[0010] Moreover, examination of logic logic edit [in / as a system by which it is proposed other than the above-mentioned system / "structured statement document]": Eiji Yamazaki, Chizuko Suzuka, Takako Sugiyama, the 39th time (common Narimoto year anaphase) national conference of Information Processing Society of Japan.

What was indicated by ***** is known.

[0011] He takes in the subordination child generation child defined by Above ODA as constraint of structure generation, and is trying to show structure generable by evaluating that subordination child generation child in the system indicated by this reference.

[0012] For example, when there is constraint "paragraph components have a sentence, drawing, and any one component of the table in low order", the constraint is expressed with structure (this is equivalent to the common logical structure) as shown in drawing 39 . The candidate of components is chosen in order of a "sentence", "drawing", and a "table" by the constraint shown with this structure. Therefore, a system will generate sentence components first and will show them to a user. structure as a system shows now to drawing 40 (a) -- showing -- on the other hand, a user -- the sentence components of "a paragraph 3" -- specifying -- "-- others -- if demand directions of candidate" are performed, as the above-mentioned constraint is evaluated and it is shown in drawing 40 (b), a system will delete sentence components, will newly generate drawing components (candidate 2), and will show them to a user.

[0013] Desired document structure (logical structure) can be acquired by repeating such actuation.

[0014]

[Problem(s) to be Solved by the Invention] However, if the constraint of a predetermined document is given, since sequential assessment will be carried out in process of document generation of the condition, when choosing the candidate of a construct (components) and becoming generable, a subordination child production rule (constraint) is estimated for the first time by the above-mentioned conventional structured statement document editor, and a generable construct is shown by it. For this reason, the actual condition is what kind of construct being able to generate and not understanding and being unable to hold the overview of generable document structure at all at the time of edit initiation until a construct is displayed.

[0015] For example, the case where a document with the following constraint is drawn up is considered.

A <constraint 1> document consists of a list of the repeat of one foreword and chapter.

A <constraint 2> chapter consists of a list of a paragraph, a paragraph, or the option of iteration of alternative selection of reference.

A <constraint 3> foreword consists of a list of a paragraph.

[0016] This document surely consists of one foreword and one or more chapters, and neither a paragraph nor reference cannot exist in the same hierarchy as a chapter. A foreword consists of only one or more paragraphs, and must not contain any components other than a paragraph. The chapter surely contains one or more paragraphs first, after that, it is the repeat of a paragraph and reference and reference must not appear at the head of a chapter.

[0017] Now, suppose that the document with one chapter which becomes one foreword which consists of one paragraph from one paragraph was given as a document which fills the above-mentioned constraint. Namely, the "1 step of ***** failure in an examination Chapter 1 Chapter 1 1st paragraph "

Suppose that ***** was given.

[0018] If it is going to add the component part of the hierarchy same behind "***** 1 paragraph", constraint 3 will be evaluated and it will be shown that a "paragraph" can be generated. If it is going to add the component part of the hierarchy same behind "Chapter 1 The 1st paragraph", constraint 2 will be evaluated and it will be shown that a "paragraph" or "reference" can be generated. If it is going to add the component part of the hierarchy same before "Chapter 1 The 1st paragraph" similarly, it will be shown that constraint 2 is evaluated, and a "paragraph" can be generated since the head of a "chapter" must be a "paragraph." Here, it does not turn out that "reference" cannot be added before "Chapter 1 The 1st paragraph" until it is going to add a component part before "the Chapter 1 1st paragraph" actually.

[0019] Moreover, when a certain hierarchy's component part is generated, it is not known what kind of structure may be further generated by the low order until it evaluates a low-ranking constraint. If it is going to add this hierarchy's component part behind "Chapter 1", constraint 1 will be evaluated and it will be shown that a "chapter" can be generated. "Chapter 2" will be generated if a "chapter" is generated according to presentation.

[0020] Therefore, "the Chapter 1 Chapter 1 step [1 step of ***** failure-in-an-examination / 1st step] failure in an examination of Chapter 2 "

The document to say is obtained.

[0021] It is shown that a component part generable in the newly generated low order of "Chapter 2" at this time tends to generate a component part in low order actually, constraint 2 is evaluated, and a "paragraph" can be generated. Moreover, if it is going to generate this hierarchy's component part behind the Chapter 1 1st paragraph, constraint 2 will be evaluated and it will be shown that a "paragraph" and "reference" can be generated. If it is going to generate "reference" behind "***** 1 paragraph" similarly, constraint 3 is evaluated, it is shown that only a "paragraph" can be generated and it cannot generate "reference."

[0022] Therefore, " ***** 1 paragraph x referring-to-x (it is ungenerable)

The Chapter 1 Chapter 1 1st paragraph reference chapter 2 1st paragraph of Chapter 2 "

The document to say is obtained.

[0023] Thus, in the conventional structured statement document editor mentioned above, a generable component part (structure) is unknown until it generates actually. For this reason, when generation of desired structure is not completed, it goes back to the choosing point of a high order, a substructure (structure generated until now after a choosing point) is canceled, and new structure (candidate) must be rechosen. Moreover, in order to cancel unnecessary structure since the constraint for structure generation cannot be seen from a user and only the structure in which the generation as an assessment result of constraint is possible is shown, and to acquire new structure, there are no means which know to which choosing point it should go back, and trial-and-error must be repeated. Therefore, the great effort was needed for generating desired structure (specific structure).

[0024] This invention aims at offering the document processing system equipment which enables generation and edit of the document structure which a user desires in the range with which are satisfied of the constraint which common structure has while it displays the whole picture of the candidate of the structure generable at the time of generation and edit of a structured statement document and enables selection of the candidate of arbitration.

[0025]

[Means for Solving the Problem] In order to attain the above-mentioned object, based on the constraint imposed on logical structure information common to two or more documents in this invention, the logical structure information on a specific document is interactively set to the document processing system equipment in which generation and edit are possible. While generating the candidate of the logical structure information on all generable specific documents who satisfies the constraint which shows said constraint and showing with the constraint concerned, it has a structure generation means to generate and show one logical structure information according to a demand out of the shown candidate.

[0026]

[Function] According to the document processing system equipment concerning this invention, a structure generation means generates the candidate of all the specific structures in which the generation with which are satisfied of the constraint (for example, subordination child generation child) which shows the constraint imposed on the common structure which is logical structure information common to two or more documents is possible, and presents him with a constraint. A user does a directions demand that a desired construct should be generated out of the construct candidate group which constitutes the candidate of each specific structure with reference to the candidate and constraint of all specific structures which were shown. Then, according to the directions demand from a user, a structure generation means generates and presents specific structure so that it may be satisfied with a constraint.

[0027] Therefore, according to this invention, since the generable candidate and generable constraint of all specific structures were shown, a user can create easily the specific structure of the document for which it asks.

[0028]

[Example] Hereafter, the example of this invention is explained with reference to drawing 1 thru/or drawing 35.

[0029] Drawing 1 shows one example of the document processing system equipment concerning this invention by the functional block diagram.

[0030] In this drawing, the common structure storing section 110 stores common structure (common logical structure).

[0031] A structure generation means 120 reads the common logical structure from the common structure storing section 110, generates the candidate (namely, the construct candidate group which can become, each element, i.e., each construct, which constitutes the logical structure) of the logical structure information on all generable specific documents that the constraint is satisfied while reading the constraint which shows the constraint on the structure at the time of generating specific structure (specific-logical structure) from the common logical structure, and presents him with the constraint concerned. Moreover, one logical structure information according to a demand is generated and shown out of the shown candidate.

[0032] The specific structure construction section 130 consists of the construct generation section 131 and the specific structure storing section 132, and builds specific structure based on the generated construct. The construct generation section 131 stores in the specific structure storing section 132 the construct generated by the structure generation means 120. The construct newly generated by the specific structure already generated at this time is connected.

[0033] The generation directions section 140 notifies directions information, such as a demand, to the structure generation means 120 that the construct inputted when a user operated the designating devices 150, such as a mouse, should be generated.

[0034] A display and control section 160 carries out the display control of the information, such as candidate structure generated by the structure generation means 120, and specific structure (the structure in a generation process is also included) set as the object of the processing currently held at the specific structure storing section 132, to the displays 170, such as a display.

[0035] In addition, in this example, the common structure storing section 110 and the specific structure storing section 132 are realized by storage, such as a hard disk and main memory, and the generation directions section 140 and a display and control section 160 are realized by the control means of a processor, a central processing unit, etc., and further, the structure generation means 120 and the construct generation section 131 are realized, when the above-mentioned control means performs software (program) for making each function of a means 120 and the generation section 131 carry out. Of course, it is also realizable with hardware and firmware.

[0036] Moreover, in this example, the structure of a document is treated in accordance with the agreement specified by Above ODA. In Above ODA, the subordination child production rule which is constraint to specific structure from common structure is stored as a subordination child generation child who is the attribute of the construct which constitutes common structure. The subordination child production rule stored as a subordination child generation child is expressing the construct which can be subordinate to the low order of a construct with the subordination child generation child under the direct control by the structure generation formula (construction expression) which specifies the combination of the construct. Since there are SEQ, REP, OPT, CHO, and OPTREP in the formation rule which constitute the structure generation type, those regulations are explained below.

[0037] ◇ SEQ (A, B, C--) is A, B, and C. -- A list is shown and they are A, B, and C. -- All are indispensable and appearance order watch is immobilization.

◇ REP (A) shows the repeat of a 1-time or more arbitration time of A.

◇ OPT(A) shows the option of A, and even if A appears, it is not necessary to carry out it. (However, an appearance is ** at a time)

◇ CHO (A, B, C--) shows the alternative selection from A, B, and C--, and some can appear one among A, B, and C.

◇ OPTREP (A) shows the repeat of a 0 times or more arbitration time of A.

[0038] A document consists of a repeat of a chapter here. -- Constraint of Chapter 1 consists of a list of a paragraph, a paragraph, or the option of iteration of alternative selection of reference. -- It is constraint called constraint 2 <constraint 1>. Doc : := REP (Section)

<Constraint> 2 Section :: = SEQ (Paragraph and OPTREP (CHO (Paragraph, Reference))) (1)

It can express by the structure generation formula to say.

[0039] While generating the minimum candidate structure where the structure generation means 120 satisfies read-out and its regulation for a subordination child production rule (structure generation type) from common structure as pretreatment, on the occasion of creation of the specific structure which starts this invention here and showing, the minimum structure is generated and shown by evaluating said subordination child production rule (structure generation type). The information on the generated minimum candidate structure and the minimum structure is inputted and displayed on a display 170 through a display and control section 160. In addition, about the detail of generation of the minimum candidate structure and the minimum structure, it mentions later.

[0040] In this example, in the case of creation and editing operation of specific structure, as shown in drawing 2, the window 200 which has the candidate structure viewing area 210 for displaying candidate structure and the specific structure viewing area 220 for displaying the specific structure corresponding to the candidate structure currently displayed on the candidate structure viewing area 210 is displayed on the display screen of a display 170.

[0041] In drawing 2, minimum candidate structure 210A which satisfies the structure generation type of the above-mentioned formula (1) is displayed on the candidate structure viewing area 210, and minimum structure 220A which evaluated the structure generation type of the above-mentioned formula (1), and was obtained is displayed on the specific structure viewing area 220. Incidentally, the formation rule of "REP" in minimum candidate structure 210A and "SEQ" are evaluated, and minimum structure 220A is obtained.

[0042] In addition, each element which constitutes candidate structure (structure where new structure was added to the minimum candidate structure) including the minimum candidate structure from this example is defined as a node. As shown by each node which constitutes minimum candidate structure 210A among these nodes The node equivalent to the roots (root), such as "Logical Root Class", is defined as a root node. Moreover, the node showing the candidate who can become constructs, such as "Section", two "Paragraph(s)", and "Reference", is defined as a candidate node. Furthermore, the node showing formation rules, such as "REP", "SEQ", "OPTREP", and "CHO", is defined as a GFS (Generator for subordinates) node.

[0043] By the way, since the structure below the node can newly be added to minimum candidate structure 210A shown in drawing 2 by carrying out multiple-times application (multiple-times assessment) of the addition of structure, for example, the GFS node of "REP" and "OPTREP", the minimum candidate structure is not always displayed on the candidate structure viewing area 210. Since the construct corresponding to the candidate node "Paragraph" which similarly exists in the low order of the node by evaluating the GFS node in minimum candidate structure 210A "OPTREP" is generated and the construct is connected to the low order of the construct in minimum structure 220A "aSection", the minimum structure is not always displayed on the specific structure viewing area 220.

[0044] Moreover, in this example, the structure which a certain GFS node has as a low order node is defined as a candidate substructure. For example, in minimum candidate structure 210A, the structure in the dotted line shown with the sign 230 in drawing 2 which the GFS node "SEQ" has as a low order node is a candidate substructure, and the structure in the dotted line shown with the sign 240 in drawing 2 which the GFS node "OPTREP" has as a low order node is a candidate substructure similarly.

[0045] Furthermore, what was generated corresponding to the candidate substructure is defined as the substructure in specific structure. For example, in the candidate substructure 230, if the GFS node "OPTREP" is not chosen, the construct corresponding to a candidate node "Paragraph" serves as a substructure, and if the candidate node "Paragraph" is chosen by the regulation of a GFS node "CHO", in the candidate substructure 240, the construct corresponding to a candidate node "Paragraph" will serve as a substructure similarly. Moreover, in the candidate substructure 230, if a GFS node "OPTREP" is chosen and the candidate node "Reference" is chosen by the regulation of a GFS node "CHO", the structure which consists of each construct corresponding to a candidate node "Paragraph" and "Reference" will turn into a substructure.

[0046] Next, specific structure generation / edit processing by the structure generation means 120 is explained with reference to drawing 3 thru/or drawing 8.

[0047] In addition, the flow chart generation processing actuation of low-order structure is shown [flow chart / generation processing actuation of the minimum structure is shown / flow chart / expansion processing actuation of a structure generation type is shown / flow chart / expansion processing actuation of a subordination child generation child is shown / flow chart / generation processing actuation of the minimum candidate structure is shown / flow chart / the processing actuation of the whole specific structure generation / edit processing is shown in drawing 3 / in drawing 4 / in drawing 5 / in drawing 6 / in drawing 7] in drawing 8 is shown, respectively.

[0048] As shown in drawing 3, the subroutine for the minimum candidate structure generation processing is called, and the structure generation means 120 generates the minimum candidate structure from constraint of the read common structure while reading common structure from the common structure storing section 110 (step 310) (step 320). Then, the minimum candidate structure is sent out and it is made to display on a display 170 to a display and control section 160 (step 330). Next, while calling the subroutine for the minimum structure generation processing and generating the minimum candidate structure to the generated minimum structure (step 340), the minimum structure is sent out and it is made to display on a display 170 to a display and control section 160 (step 350). Then, when judging whether it is the no which editing operation ended (step 360), ending processing when it ends, and having not ended on the other hand While reading the subroutine for edit directions demand processing and generating structure (a candidate substructure and substructure) according to the directions from the generation directions section 140 corresponding to the editing operation from a user (step 370) According to generation of the structure, the display of candidate structure and specific structure is updated (step 380). After step 380 termination returns to the above-mentioned step 360, and performs this step or subsequent ones.

[0049] Next, the minimum candidate structure generation processing is explained with reference to drawing 4.

[0050] If the subroutine for the minimum candidate structure generation processing is called, the structure generation means 120 will call the subroutine for attribute GFS expansion processing, and will perform expansion processing of an attribute GFS while reading the subordination child generation child as an attribute GFS (Generator forsubordinates) of the logic root (Root) of a document (step 410), as shown in drawing 4 (step 420). If step 420 is ended, a return will be carried out to step 320 of drawing 3.

[0051] Next, attribute GFS expansion processing is explained with reference to drawing 5

[0052] If the subroutine for attribute GFS expansion processing is called, as the structure generation means 120 is shown in drawing 5, it will judge whether the formula of "GFS=nil" (GFS is not defined) is materialized (step 510). Here, when the attribute GFS is defined (the formula of the above "GFS=nil" is abortive), the subroutine for structure generation type expansion processing is called, and the structure generation type (construction expression:CE and brief sketch) of GFS is developed (step 520). When step 520 is ended and the attribute GFS is not defined by the above-mentioned step 510, a return is carried out to step 420 of drawing 4.

[0053] Next, structure generation type expansion processing of an attribute GFS is explained with reference to drawing 6 .

[0054] If the subroutine for structure generation type expansion processing is called, the structure generation means 120 will read the argument of a structure generation type (step 620), and will take out a top argument while it generates the GFS node (GFS-nodo) corresponding to the formation rule of a structure generation type and connects it as a child of a host node (high order nodo) (step 610), as shown in drawing 6 (step 630).

[0055] Next, it judges whether it is the no in which the formula of argument =CE (structure generation type) is materialized (step 640).

[0056] Here, when the above-mentioned formula is abortive (i.e., when an argument is a construct), while generating the candidate node (candidate-nodo) corresponding to a construct and tying as a child of a GFS node (step 650), the subroutine for attribute GFS expansion processing (refer to drawing 5) is called recursively, and the attribute GFS of the construct is developed (step 660).

[0057] On the other hand, when the formula of "argument =CE" is materialized at step 640 (i.e., when an argument is a structure generation type), the subroutine for structure generation type expansion processing is called recursively, and a structure generation type is developed by making the generated GFS node into a host node (step 670).

[0058] If the above-mentioned step 660 or step 670 is ended, when it judges whether all the arguments of the structure generation type read in the above-mentioned step 620 were evaluated (step 680) and there is a non-evaluated argument, the structure generation means 120 returns the following argument to ejection (step 690) and its account step 640 of Gokami, and performs this step or subsequent ones, and when all are evaluated, on the other hand, it will carry out a return to step 520 of drawing 5 .

[0059] Next, the minimum structure generation processing is explained with reference to drawing 7 .

[0060] If the subroutine for the minimum structure generation processing is called, as the structure generation means 120 is shown in drawing 7 , the construct corresponding to a root node (Root node) will be generated (step 710), and it will judge whether a low order node exists in the construct (step 720).

[0061] Here, when the low order node exists, the subroutine for low order structure generation processing is called, a low order node is taken out from the construct generated in the above-mentioned step 710 after that, and low order structure is generated (step 730). And when step 730 is ended and a low order node does not exist at the above-mentioned step 720, a return is carried out to step 340 of drawing 3 .

[0062] Next, low order structure generation processing is explained with reference to drawing 8 .

[0063] If the subroutine for low order structure generation processing is called, as shown in drawing 8 , the structure generation means 120 will judge whether it is a GFS node (step 801), and will perform processing (assessment of a node) according to a decision result.

[0064] Here, when it is a candidate node, a corresponding construct is generated and the this generated construct is connected recently as a child of the construct generated by the candidate node of a high order.

[0065] On the other hand, when it is a GFS node, the following processings are performed according to the structure generation type of GFS.

[0066] \diamond SEQ : evaluate altogether the node of low order under the direct control sequentially from a head.

\diamond REP : evaluate the node of low order under the direct control once.

\diamond OPT : close assessment.

\diamond CHO : evaluate only the head of the node of low order under the direct control.

\diamond OPTREP : close assessment.

[0067] Then, in the above-mentioned step 801, it judges whether in the case of a GFS node, it is "CE=SEQ" (step 802), and in SEQ, the subroutine for low order structure generation processing is called for the head of a low order node ejection (step 803) and after that, and it generates low order structure here (step 804). Next, it judges whether all the low order node was processed (step 805), when there is an unsettled low order node, the following low order node is taken out (step 806), and it returns to the above-mentioned step 804, and this step or subsequent ones is performed.

[0068] Judging whether in "CE!=SEQ", it is "CE=REP" at step 802, (step 807) in REP, the subroutine for low order structure generation processing is called for a low order node ejection (step 808) and after that, and it generates low order structure here (step 809).

[0069] Judging whether in "CE!=REP", it is "CE=CHO" at step 807, (step 810) in CHO, the subroutine for low order structure generation processing is called for the head of a low order node ejection (step 811) and after that, and it generates low order structure here (step 812).

[0070] In "CE!=CHO", it judges whether it is "CE=OPT" at step 810 (step 813), and, also in [no] OPT, carries out here.

[0071] In "CE!=OPT", it judges whether it is "CE=OPTREP" at the above-mentioned step 813 (step 814), and, in no cases of OPTREP, carries out here.

[0072] When it is not a GFS node in the above-mentioned step 801 that is, in the case of a candidate node, the construct corresponding to a candidate node is generated, and it connects the construct as a child of the construct generated from the candidate node of the latest high order (step 815). Next, when it judges whether a low order node exists in the construct (step 816) and a low order node exists here, the subroutine for low order structure generation processing is called for a low order node ejection (step 817) and after that, and low order structure is generated (step 818).

[0073] In addition, when all the low order node is processed in the above-mentioned step 805, the above-mentioned step 809, step 812, or step 818 is ended and a low order node does not exist in the above-mentioned step 816 at the above-mentioned step 813 or step 814 in "YES", the return of the case of ***** is carried out to step 730 of drawing 7.

[0074] In addition, about the DS holding candidate structure, it is realizable by using the DS (generator) currently indicated by Japanese Patent Application No. No. 157067 [two to] for which the applicant for this patent applied previously. Moreover, generation of the minimum structure is realizable by using the approach of the minimum structure generation of allocation good Atagi currently indicated by Japanese Patent Application No. No. 157068 [two to] for which the applicant for this patent applied previously.

[0075] It means that the minimum candidate structure and the minimum structure were generated by the above processing.

[0076] And if the selection information is inputted into the structure generation means 120 through the generation directions section 141 by a user's operating a designating device 150 and choosing a GFS node, the generation means 120 sends out the candidate of structure generation processing according to the formation rule of a GFS node based on the inputted selection information to a display and control section 160. Then, a display and control section 160 displays the candidate of structure generation processing who inputted on a display 170.

[0077] Here, the structure generation processing according to the formation rule of the attribute GFS defined in this example it can be [****] adapted is explained for every formation rule.

[0078] \diamond nothing in SEQ -- in \diamond REP, there are processing [add subordinate] and processing [delete subordinate]. [add subordinate] is the addition of a subordinate node. It adds to the candidate structure already generated by making the candidate substructure (candidate structure which developed the node chosen as a root) which it has as a low order node into the youngest child, and adds to the specific structure which generates the substructure (structure which generated the node chosen as a root) corresponding to a candidate substructure further, and has already been generated. [delete subordinate] is deletion of a subordinate node. When two or more nodes are generated as a low order node, the last node is deleted from candidate structure and the construct corresponding to the node is deleted from the specific structure already generated.

[0079] \diamond There are processing [select] and processing [reject] in OPT. [select] chooses a low-ranking node. It adds to the specific structure which generates the substructure corresponding to the candidate substructure which it has as a low-ranking node, and has already been generated.

[reject] -- Un-choosing [of a low-ranking node].

The substructure corresponding to the candidate substructure which it has as a low-ranking node is deleted from the candidate structure already generated.

[0080] \diamond There is processing [n-subordinate] in CHO. [n-subordinate] chooses the n-th low order node. The substructure corresponding to the candidate substructure which it has as a node of the low order chosen before is deleted from the specific structure already generated, the substructure corresponding to the newly chosen node is generated, and it adds to the specific structure.

[0081] \diamond In OPTREP, there are processing [select], processing [reject], processing [add subordinate], and processing [delete subordinate]. [select] and [reject] -- the case of OPT -- the same. [add subordinate] is the addition of a subordinate node. Although this is the

same as that of the case of REP, it acts, only when the low order node is chosen. [delete subordinate] is deletion of a subordinate node. Although this is the same as that of the case of REP, it acts, only when the low order node is chosen.

[0082] Next, processing of the edit directions demand to the minimum candidate structure mentioned above is explained with reference to drawing 9 .

[0083] As shown in drawing 9 , it asks for the GFS node by which the edit demand from a user was emitted, the formation rule which the GFS node has are investigated (step 901), and processing according to the result is performed.

[0084] In the above-mentioned step 901, it investigates whether it is "CE=REP" (step 902), and, in REP, they are the list [add subordinate] of edit processing applicable here, and [delete subordinate].

A user is shown (step 903), the subroutine for editing operation processing in a GFS node [REP] is called further, and a construct is generated according to directions of a user (step 904).

[0085] It investigates whether it is "CE=CHO" when it is not REP in the above-mentioned step 902 (step 905), and, in CHO, they are an edit list [1-subordinate] applicable here, --, [n-subordinate].

A user is shown (step 906), the subroutine for editing operation processing in a GFS node [CHO] is called further, and a construct is generated according to directions of a user (step 907).

[0086] It investigates whether it is "CE=OPT" when it is not CHO in the above-mentioned step 905 (step 908), and, in the case of OPT, is an edit list [select] and [reject] applicable here.

A user is shown (step 909), the subroutine for editing operation processing in a GFS node [OPT] is called further, and a construct is generated according to directions of a user (step 910).

[0087] It investigates whether it is "CE=OPTREP" when it is not OPT in the above-mentioned step 908 (step 911), and, in OPTREP, is an edit list [select], [reject], [add subordinate], and [delete subordinate] applicable here.

A user is shown (step 912), the subroutine for editing operation processing in a GFS node [OPTREP] is called further, and a construct is generated according to directions of a user (step 913).

[0088] In addition, when one step of the above-mentioned step 904, step 907, step 910, and step 913 is ended, a return is carried out to step 370 of drawing 3 .

[0089] Next, activation of called subroux CHIN for editing operation processing by the structure generation means 120 is explained for every GFS node.

[0090] The algorithm of the editing operation in a GFS node [REP]. As shown in drawing 10 , it is applied and edit processing is investigated (step 1001). it adds to the specific structure which investigates whether processing is [add subordinate] (step 1002), adds to the candidate structure already generated, generates the substructure corresponding to a candidate substructure further by making into the youngest child the candidate

substructure which it has as a low order node if it comes out so and is, and has already been generated (step 1003). It investigates whether processing is [delete subordinate] if it is not [add subordinate] at the above-mentioned step 1002 (step 1004), and if that is right, it will investigate whether there are two or more low order nodes (step 1005). If there are two or more here, the node of the last of a low order node will be deleted from candidate structure, and the construct corresponding to the deleted node will be deleted from specific structure (step 1006). In addition, when the above-mentioned step 1003 or step 1006 is ended and the number of low order nodes is one in the above-mentioned step 1005, a return is carried out to step 904 of drawing 9 .

[0091] The algorithm of the editing operation in a GFS node [CHO]. As shown in drawing 11 , the applied edit processing is investigated (step 1101). The substructure corresponding to the candidate substructure which it has as a node of the low order chosen before It deletes from the specific structure already generated (step 1102), the substructure corresponding to the candidate substructure which it has as a selected low order node after that is generated, and it adds to the specific structure in step 1102 (step 1103). The return of after step 1103 termination is carried out to step 907 of drawing 9 .

[0092] The algorithm of the editing operation in a GFS node [OPT]. As shown in drawing 12 ; the applied edit processing is investigated (step 1201). it investigates whether processing is [select] (step 1202), and if it comes out so and is, it will add to the candidate structure which generates the substructure corresponding to the candidate substructure which it has as a low order node, and has already been generated (step 1203). it investigates whether processing is [reject] if it is not [select] in the above-mentioned step 1202 (step 1204), and if it comes out so and is, the substructure corresponding to the candidate substructure which it has as a low order node will be deleted from the candidate structure already generated (step 1205). In addition, if the above-mentioned step 1203 or step 1205 is ended, a return will be carried out to step 910 of drawing 9 .

[0093] The algorithm of the editing operation in a GFS node [OPTREP]. As shown in drawing 13 , the applied edit processing is investigated (step 1301). it investigates whether processing is [select] (step 1302), and if it comes out so and is, it will add to the specific structure which generates the substructure corresponding to the candidate substructure which it has as a low-ranking node, and has already been generated (step 1303). it investigates whether processing is [reject] if it is not [select] at the above-mentioned step 1302 (step 1304), and if it comes out so and is, the substructure corresponding to the candidate substructure which it has as a low-ranking node will be deleted from the specific structure already generated (step 1305).

[0094] Investigate whether processing is [add subordinate] if it is not [reject] at the above-mentioned step 1304 (step 1306), and if that is right If it judges whether the node is chosen (select) (step 1307) and is chosen here (select) It adds to the candidate structure already generated by making into the youngest child the candidate substructure which it has as a low order node, and adds to the specific structure which generates the substructure corresponding to a candidate substructure further, and has already been generated (step 1308).

[0095] If it investigates whether processing is [delete subordinate] if it is not [add subordinate] at the above-mentioned step 1306 (step 1309), it judges whether the node is chosen if that is right (select) (step 1310) and it is chosen here (select), it will judge whether there are two or more low order nodes (step 1311). In a certain case, the node of the last of a low order node is deleted from candidate structure, and two or more low order nodes delete the construct corresponding to the deleted node from specific structure here (step 1312).

[0096] In addition, when one step of the above-mentioned step 1303, step 1305, step 1308, and step 1312 is ended, the node is not chosen in the above-mentioned step 1307 or step 1310 and there is only one **** low order node in the above-mentioned step 1311, a return is carried out to step 913 of drawing 9.

[0097] Next, an example is given and generation processing of the minimum candidate structure, the minimum structure, and specific structure is explained.

[0098] As constraint <constraint 1> Doc :: = REP (Section)

<Constraint> 2 Section :: = REP (SEQ (Paragraph, OPTREP (CHO (Paragraph, SEQ (Reference No, Refered No, Reference Body))))))

When the becoming structure generation formula (construction expression:CE and brief sketch) is given, the minimum candidate structure will be generated by performing processing based on the flow chart of drawing 4 mentioned above, drawing 5, and drawing 6.

[0099] Namely, DOC of the attribute GFS which is the (1) constraint 1 :: = REP (Section) is developed.

[0100] (2) A GFS node [REP] is generated, it is tied directly under Doc which is a root, and an argument [Section] is evaluated further.

[0101] (3) Since [Section] is a construct, a candidate node [Section] is generated and it is connected as a child of a GFS node [REP].

[0102] (4) Next, the constraint 2 of the attribute GFS of [Section] is developed.

[0103] (5) A GFS node [REP] is generated and it is tied as a child of a candidate node [Section].

[0104] (6) The argument of formation rule REP is evaluated, a GFS node [SEQ] is generated, and it is tied as a child of a GFS node [REP].

[0105] (7) The argument of formation rule SEQ is evaluated in order.

[0106] That is, the 1st argument [Paragraph] of formation rule SEQ is evaluated, a candidate node [Paragraph] is connected as a child of a GFS node [SEQ], and GFS of [Paragraph] is developed further. Since the [Paragraph] does not have an attribute GFS, assessment stops.

[0107] And CE[OPTREP of the 2nd argument of formation rule SEQ (CHO(Paragraph, SEQ (Reference No, Refered No, Reference Body)))]

***** is performed.

[0108] Similarly, the candidate structure 1410 will be generated as displayed on the 1st argument [Paragraph] of formation rule CHO, and candidate structure viewing-area 1400 of window 1400 shown in drawing 14 as the result when 1st argument [of formation rule SEQ] [Reference No], 2nd argument [Referred No], and 3rd argument [ReferenceBody] are evaluated further, respectively A. This candidate structure 1410 is the minimum candidate structure, and addition of structure and deletion are performed to this structure. The object (it is called Object and Following Obj) 1 (aLogical RootClass) is generated by specific structure viewing-area 1300B. In addition, OBJ1 ("Logical Root Class") currently displayed on viewing-area 1400B corresponds to "Doc" in the structure generation type (constraint 1) mentioned above.

[0109] When the minimum candidate structure is generated, the minimum structure will be generated by the processing based on the flow chart of drawing 7 and drawing 8 based on this minimum candidate structure.

[0110] That is, from the minimum candidate structure, the construct of a root (Root) node to the root is generated. Since the low order node of a root node is a GFS node of "REP", a low-ranking node is evaluated further, the construct of "Section" is generated, and it is connected as a child of the construct of the root. Since the low order node of the candidate node of "Section" is "REP", the low-ranking GFS node of "SEQ" is evaluated further, and the construct of "Body Paragraph" is connected with the construct of "Section." And assessment is closed by the GFS node of "OPTREP." The specific structure 1500 will be generated as displayed on viewing-area 1400B of the window 1400 shown in drawing 15 as this result.

[0111] Obj2 (aSection) is generated as low order of Obj1 (aLogical Root Class), and, as for the specific structure 1500, Obj3 (aBody Paragraph) is further generated as low order of Obj2.

[0112] It means that the minimum structure had been generated by the above processing from the minimum candidate structure.

[0113] Next, based on the minimum candidate structure and the minimum structure, the example of edit of the specific-logical structure with which constraint was filled is explained with reference to drawing 16 thru/or drawing 26 .

[0114] If a GFS node is chosen with a designating device and a structure generation means is started from the display condition which a user shows to drawing 15 , the structure generation means 120 will show a user the candidate of structure generation processing according to the formation rule of the GFS node chosen.

[0115] If a user chooses the GFS node 1600 (GFS node [OPTREP]) with a designating device 150 on the viewing area of a window 1400 and advances an editing operation demand by the click of a mouse button etc. as shown in drawing 16 , the list of the applicable processing (candidate of structure generation processing) corresponding to a production rule [OPTREP] will be displayed by the algorithm of drawing 9 . The display condition is shown in drawing 17 : In drawing 17 , 1700 is the list of applicable processing, for example, is the list of menu formats. By choosing the item of the request in the list of the applicable processing, processing which corresponds with the structure generation means 120 is performed.

[0116] If a user directs [Select] in the list 1700 of applicable processing now as shown in drawing 18 from the display condition of drawing 17, the minimum structure generation processing will be applied to the low order node of a GFS node [OPTREP], and a new construct (aBody Paragraph) will be generated. The display condition of the specific structure in this event is shown in drawing 19.

[0117] In drawing 19, 1900 is specific structure and this specific structure 1900 has composition which Obj4 (aBody Paragraph) was generated and was added as low order of Obj2 in the configuration of the specific structure 1500 shown in drawing 15.

[0118] Next, since the list of applicable processing corresponding to a production rule [CHO] is displayed when a user chooses the GFS node 2000 (GFS node [CHO]), as shown in drawing 20, a user chooses desired processing out of the list. The list 2100 of the applicable processing corresponding to a production rule [CHO] is displayed, and drawing 21 shows that [2-subordinate] (2nd alternative) was directed out of the list.

[0119] Thus, if a production rule [CHO] is chosen and processing [2-subordinate] is chosen, the specific structure 2200 which is displayed on viewing-area 1400B shown in drawing 22 will be acquired. In the configuration of the specific structure 1900 which showed this specific structure 2200 to drawing 19 Obj4 (aBody Paragraph) as low order of Obj2 is canceled. As low order of Obj2 It corresponds to the candidate of the construct below the newly chosen GFS node 2210 (GFS node "SEQ"). It has the composition that Obj5 (aReference No), Obj6 (aRefered No), and Obj7 (aReferenceBody) were generated, respectively, and they were connected.

[0120] While similarly a user chooses the GFS node 2300 (GFS node [REP]) as shown in drawing 23, and he performs an editing operation demand further from the display condition shown in drawing 22 If [add subordinate] is directed out of the list 2400 of applicable processing of the GFS node [REP] displayed as the result as shown in drawing 24 As low order of the GFS node 2300 (GFS node [REP]) shown in drawing 23, the 2300 or less GFS node candidate substructure (structure in the dotted line shown by the drawing 23 Nakaya mark A) is generated. The display condition of the candidate substructure and the specific structure in this event is shown in drawing 25.

[0121] In drawing 25, the structure 2510 where the structure in the dotted line shown by the drawing 23 Nakaya mark A was generated as low order of the GFS node 2300 (GFS node [REP]) in candidate structure 2500A in viewing-area 1400A is connected.

[0122] On the other hand, specific structure 2500B corresponding to the candidate structure 2500 is displayed on viewing-area 1400B. By having carried out additional generation of the candidate substructure 2510, the GFS node 2520 (GFS node [SEQ]) is evaluated by the configuration of the specific structure 2200 shown in drawing 22, a construct (aBody Paragraph) is generated as that result, and this specific structure 2500B has at it the composition that that construct was connected in the low order of Obj2 as Obj8.

[0123] And if specific structure 2500B was desired structure, specific structure 2500B becomes final specific structure, i.e., specific-logical structure.

[0124] By repeating the same actuation, various specific structures are generable. An example of the specific structure is shown in drawing 26.

[0125] In the candidate structure where candidate structure 2600A shown in drawing 26 is displayed on viewing-area 1400A of drawing 25 In the candidate structure which performed processing actuation of [add subordinate] to the GFS node 2300 (GFS node "REP") once; and then was acquired as a result In the candidate structure which performed once processing actuation of [add subordinate] to the GFS node "REP" equivalent to the GFS node 2530 (GFS node "REP") shown in drawing 25 , and was acquired further as a result It will be obtained by performing twice processing actuation of [add subordinate] to the GFS node "OPTREP" equivalent to the GFS node 2540 (GFS node "OPTREP") shown in drawing 25 . In this way, in obtained candidate structure 2600A, specific structure 2600B will be obtained by generating the candidate nodes 2601-2613.

[0126] Edit of specific structure is possible, filling constraint from common structure according to this example, as mentioned above showing the whole picture of the candidate of generable specific structure.

[0127] Furthermore, in this example, since the condition of having said how many nodes the GFS node would have in that low order can be held, easy guidance of structure edit can be performed by utilizing the information on that condition.

[0128] For example, although it is the applicable processing (edit processing) [add subordinate] by the GFS node [REP], and [delete subordinate], [delete subordinate] is effective only when a low order node is two or more pieces. Then, when there is an editing operation demand by the GFS node, the condition of the GFS node can be investigated and the edit equipment which shows a user only effective editing operation can be realized.

[0129] Each node of a GFS node is explained here.

[0130] ◇ When a REP(a) low order node is less than two pieces, only processing [add subordinate] is effective.

(b) When a low order node is two or more pieces, processing [add subordinate] and processing [delete subordinate] are effective.

[0131] ◇ Only processing [disselect] is effective when the OPT(a) low order node is chosen (select).

(b) Only processing [reject] is effective when the low order node is not chosen.

[0132] ◇ Only processing [select] is effective when the OPTREP(a) low order node is not chosen.

(b) When the low order node is chosen and a low order node is less than two pieces, processing [reje ct] and processing [add subordinate] are effective.

(c) When the low order node is select(ed) and a low order node is two or more pieces, each processing of processing [reject], processing [add subordinate], and processing [delete subordinate] is effective.

[0133] Next, signs that effective applicable processing (edit processing) is restricted according to the condition of a GFS node are shown in drawing 27 thru/or drawing 35 .

[0134] The minimum candidate structure and the minimum structure are shown in drawing 27 .

[0135] As shown in drawing 27 , the minimum candidate structure 2710 is displayed on candidate structure viewing-area 2700A of a window 2700, and the minimum structure 2720 is displayed on specific structure viewing-area 2700B.

[0136] In addition, in the minimum candidate structure 2710, "Doc", "AAA", "BBB", "CCC", "DDD", "EEE", and "FFF" are candidate nodes, and "REP", two "SEQ(s)", "OPT", "OPTREP", and "CHO" are GFS nodes. On the other hand, in the minimum structure 2620, "aDoc" is a logic root and "aAAA" and "aBBB" are constructs.

[0137] If a user chooses "OPT" which is one of the GFS nodes from the display condition shown in drawing 27 as shown in drawing 28 , and he performs an edit processing demand further, the list 2900 of the applicable processing shown in drawing 29 will be displayed. Since the low order node of a GFS node "OPT" was not chosen, as for this, "select" was displayed as effective processing.

[0138] Next, if a user directs "select" in list 2900 as shown in drawing 30 , a GFS node "OPT" will be evaluated. The specific structure 3100 currently displayed on viewing-area 2700B shown in drawing 31 as the result is acquired.

[0139] Since a GFS node "OPT" is evaluated on the occasion of generation of the specific structure 3100 according to processing "select", as a substructure, a construct "aDDD" will be generated as a construct "aCCC" and its low order, and when the substructure leads to a construct "aAAA", the specific structure 3100 will be generated further.

[0140] If a user chooses a GFS node "OPT" and performs an edit processing demand again from the display condition shown in drawing 31 , since each candidate node of "CCC" and "DDD" is chosen as a low order node of a GFS node "OPT", as shortly shown in drawing 31 , the list 3200 of applicable processing in which [reject] is displayed as effective processing is displayed.

[0141] Moreover, if a user chooses a GFS node "OPTREP" and performs an edit processing demand from the display condition shown in drawing 31 , since the low order node of the GFS node "OPTREP" is not chosen, the list of applicable processing in which "select" is displayed as effective processing is displayed. And as shown in drawing 33 , when "select" in the list 3300 of the applicable processing is directed, a GFS node "OPTREP" will be evaluated and the specific structure 3400 currently displayed on viewing-area 2700B shown in drawing 34 as the result will be acquired.

[0142] In addition, drawing 34 shows the condition that 1st edit processing actuation to a GFS node "OPTREP" was performed, the construct "aEEE" was generated as a substructure, editing operation which is the 2nd [further] time was performed, and the list 3410 of applicable processing was displayed.

[0143] A candidate node "EEE" is chosen as a low order node, and since a low order node is less than two pieces, processing of "add subordinate" and [reject] is displayed on the list 3410 of applicable processing. In this case, in spite of having prepared four edit processings in the GFS node "OPTREP", edit processing was restricted to two by the regulation mentioned above.

[0144] Next, if a user directs "add subordinate" out of the list 3410 of applicable processing from the display condition shown in drawing 34 The candidate substructure which it has as a low order node (that is, by making CHO ("EEE", "FFF") into the youngest child, while adding to the specific structure already generated, the substructure (that is, in this case "aEEE") corresponding to a candidate substructure is generated.)

[0145] Similarly, when the edit processing actuation of "addsubordinate" to a GFS node "OPTREP" is repeated 3 times, the specific structure 3500 currently displayed on viewing-area 2700B shown in drawing 35 will be acquired as a result. In addition, in viewing-area 2700B of drawing 35, the construct "aEEE" which the construct "aEEE" shown by the arrow head A is generated by processing actuation (1st actuation) of "select" to a GFS node "OPTREP", and is shown by the arrow head B, and the construct "aEEE" shown by the arrow head C are generated by processing actuation (the 2nd time, 3rd actuation) of "addsubordinate" to a GFS node "OPTREP", respectively.

[0146] In addition, the construct "aEEE" which 3rd edit processing actuation to a GFS node "OPTREP" is performed, and is shown by the arrow head A as a substructure is generated, and drawing 35 shows the condition that editing operation which is the 4th [further] time was performed, and the list 3510 of edit processing was displayed.

[0147]

[Effect of the Invention] As explained above, according to this invention, a structure generation means While generating the candidate of the logical structure information on all generable specific documents who satisfies the constraint which shows the constraint imposed on logical structure information common to two or more documents and showing with the constraint concerned Since one logical structure information according to a demand is generated and it was made to show out of the shown candidate, a user While being able to know the whole picture of the candidate of the generable structure, the candidate of arbitration can be chosen at the time of generation and edit of a structured statement document, and generation and edit of desired document structure can be performed at it.

DESCRIPTION OF DRAWINGS

[Brief Description of the Drawings]

[Drawing 1] The functional block diagram showing one example of the document processing system equipment concerning this invention.

[Drawing 2] Drawing for explaining generation of the specific structure concerning this invention.

[Drawing 3] The flow chart which shows processing actuation of whole specific structure generation and edit of this example.

[Drawing 4] The flow chart which shows generation processing actuation of the minimum candidate structure of this example.

[Drawing 5] The flow chart which shows expansion processing actuation of the subordination child generation child of this example.

[Drawing 6] The flow chart which shows expansion processing actuation of the structure generation type of this example.

[Drawing 7] The flow chart which shows generation processing actuation of the minimum structure of this example.

[Drawing 8] The flow chart which shows generation processing actuation of the low order structure of this example.

[Drawing 9] The flow chart which shows processing actuation of an edit directions demand of this example.

[Drawing 10] The flow chart which shows processing actuation of the editing operation in the subordination child generation child "REP" of this example.

[Drawing 11] The flow chart which shows processing actuation of the editing operation in the subordination child generation child "CHO" of this example.

[Drawing 12] The flow chart which shows processing actuation of the editing operation in the subordination child generation child "OPT" of this example.

[Drawing 13] The flow chart which shows processing actuation of the editing operation in the subordination child generation child "OPTREP" of this example.

[Drawing 14] Drawing for explaining the minimum candidate structure in this example.

[Drawing 15] Drawing for explaining the minimum structure in this example.

[Drawing 16] Drawing for explaining the generation process for acquiring specific structure from the minimum candidate structure in this example.

[Drawing 17] Drawing for explaining the generation process for acquiring specific structure from the minimum candidate structure in this example.

[Drawing 18] Drawing for explaining the generation process for acquiring specific structure from the minimum candidate structure in this example.

[Drawing 19] Drawing for explaining the generation process for acquiring specific structure from the minimum candidate structure in this example.

[Drawing 20] Drawing for explaining the generation process for acquiring specific structure from the minimum candidate structure in this example.

[Drawing 21] Drawing for explaining the generation process for acquiring specific structure from the minimum candidate structure in this example.

[Drawing 22] Drawing for explaining the generation process for acquiring specific structure from the minimum candidate structure in this example.

[Drawing 23] Drawing for explaining the generation process for acquiring specific structure from the minimum candidate structure in this example.

[Drawing 24] Drawing for explaining the generation process for acquiring specific structure from the minimum candidate structure in this example.

[Drawing 25] Drawing for explaining the generation process for acquiring specific structure from the minimum candidate structure in this example.

[Drawing 26] Drawing for explaining the generation process for acquiring specific structure from the minimum candidate structure in this example.

[Drawing 27] Drawing for explaining signs that effective edit processing is restricted by the subordination child generation child node in this example.

[Drawing 28] Drawing for explaining signs that effective edit processing is restricted by the subordination child generation child node in this example.

[Drawing 29] Drawing for explaining signs that effective edit processing is restricted by the subordination child generation child node in this example.

[Drawing 30] Drawing for explaining signs that effective edit processing is restricted by the subordination child generation child node in this example.

[Drawing 31] Drawing for explaining signs that effective edit processing is restricted by the subordination child generation child node in this example.

[Drawing 32] Drawing for explaining signs that effective edit processing is restricted by the subordination child generation child node in this example.

[Drawing 33] Drawing for explaining signs that effective edit processing is restricted by the subordination child generation child node in this example.

[Drawing 34] Drawing for explaining signs that effective edit processing is restricted by the subordination child generation child node in this example.

[Drawing 35] Drawing for explaining signs that effective edit processing is restricted by the subordination child generation child node in this example.

[Drawing 36] Drawing for explaining edit of the specific structure (document) in the former.

[Drawing 37] Drawing for explaining edit of the specific structure (document) in the former.

[Drawing 38] Drawing for explaining edit of the specific structure (document) in the former.

[Drawing 39] Drawing for explaining edit of the specific structure (document) in the former.

[Drawing 40] Drawing for explaining edit of the specific structure (document) in the former.

[Description of Notations]

110 [-- The construct generation section, 132 / -- The specific structure storing section,
140 / -- The generation directions section, 150 / -- A designating device, 160 / -- A
display and control section, 170 / -- 1400 A display, 2700 / -- A window, 1400A, 2700A /
-- A candidate structure viewing area, 1400B, 2700B / -- Specific structure viewing area.
] -- The common structure storing section, 120 -- A structure generation means, 130 --
The specific structure construction section, 131

[Translation done.]

fchs_ws #1632220 v1

Japanese Patent Laid-Open No. 5-108631

Partial English translation of Japanese Patent Laid-
Open No. 5-108631

(43) Laid-opened Date: 30.04.1993

5 (54) Title of the invention:

DOCUMENT PROCESSOR

(21) Application Number: 03-267905

(22) Filing Date: 16.10.1991

(71) Applicant: FUJI XEROX CO LTD

10 (72) Inventor: SUZUKI KATSUAKI

(57) ABSTRACT

PURPOSE: To select an arbitrary candidate, and also, to generate and edit a user's desired document structure within a range for satisfying a constraint which a common structure has by displaying the whole contents of a candidate of a structure which can be generated at the time of generating and editing a structured document. CONSTITUTION: A structure generating means 120 reads out a common structure being logical structure information being common to plural documents, from a common structure store part 110, and also, generates candidates of all specific structures which can be generated, for satisfying a constraint condition for showing a constraint imposed to its common structure, and presents them together with the constraint.

15

20

25

Japanese Patent Laid-Open No. 5-108631

condition. Its candidate and constraint condition are displayed on a display device 170 through a display control part 160. A generation instructing part 140 informs an instruction request to the structure

5 generating means 120 in order to generate a desired structure body from in a structure body group for constituting each specific structure, from a user. The structure generating means 120 generates and presents a specific logical structure so as to satisfy the

10 constraint condition in accordance with the instruction request from the user inputted through the generation instructing part 140, and also, stores it in a specific structure store part 132

Japanese Patent Laid-Open No. 5-108631

[0005]

For example, as shown in Figure 36, in a chapter 2 composed of four sections of a "section 2.1", a "section 2.2", a "section 2.3" and a "section 2.4", if an operation of additionally inserting a part is performed with attention to the section 2.4 (chapter 2, section 4), [a prior section part, a subsequent lower section part, a subsequent section part, and a subsequent chapter part] are presented as candidates which can be additionally inserted. In Figure 36, these candidates show that, for example, "2.4-", "2.4.1", "2.4+" and "2+" can be additionally inserted as the prior section part, the subsequent lower section part, the subsequent section part, and the subsequent chapter part, respectively.

[0006]

Moreover, if the section 2.3 is paid attention to, [the prior section part, the subsequent lower section part, and the subsequent section part] are presented as the candidates which can be additionally inserted. In Figure 36, these candidates show that "2.3-", "2.3.1" and "2.3+" can be additionally inserted as the prior section part, the subsequent lower section part, and the subsequent section part, respectively. It should be noted that if the "section 2.3" is paid attention to, since subsequently there is the section 2.4, division of the chapter (chapter 2) by inserting the chapter

Japanese Patent Laid-Open No. 5-108631

part is prohibited.

[0007]

Figures 37 and 38 show display examples of the candidates which can be additionally inserted as described above.

[0008]

In the example shown in Figure 37, in the chapter 2 (2 Second Section) composed of four sections of "section 2.1 CCCCC", "section 2.2 DDDDD", "section 2.3 EEEEE" and "section 2.4 FFFFF" displayed in a window 3700 displayed on a display screen, as a result of performing the operation of additionally inserting the part with respect to the "section 2.4", a list of the candidates which can be additionally inserted is displayed in a window 3710. In the candidates displayed in the window 3710, "Within selection" means that the subsequent lower section part can be additionally inserted, "Section level 2 before" means that the prior section part can be additionally inserted, "Section level 2 after" means that the subsequent section part can be additionally inserted, and "Section level 1 after" means that the subsequent chapter and section parts can be additionally inserted, respectively.

[0009]

Also in the example shown in Figure 38, as a result of performing the operation of additionally

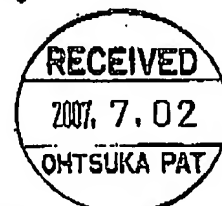
Japanese Patent Laid-Open No. 5-108631

inserting the part with respect to the "section 2.3" in the chapter 2 shown in Figure 37, a list of the candidates which can be additionally inserted is displayed in a window 3810.

整理番号:250223 発送番号:322197 発送日:平成19年 7月 2日 1

拒絶理由通知書

特許出願の番号	特願2002-371003
起案日	平成19年 6月26日
特許庁審査官	浜岸 広明 9845 5M00
特許出願人代理人	大塚 康徳(外 3名) 様
適用条文	第29条柱書、第29条第2項



この出願は、次の理由によって拒絶をすべきものである。これについて意見があれば、この通知書の発送の日から60日以内に意見書を提出して下さい。

理 由

理由1:

この出願の下記の請求項に係る発明は、下記の点で特許法第29条第1項柱書に規定する要件を満たしていないので、特許を受けることができない。

記

請求項1-4には、文書処理方法における一連の動作ステップが記載されているが、各動作ステップの主体が明示されておらず、人間が各動作を行うものをも含む記載となっている。そのため、上記請求項に係る発明は人間の精神活動に基づいて行われる処理を含んでいるものと認められる。

また、仮に各動作ステップの主体がコンピュータである場合でも、当該コンピュータが備えるハードウェア資源が具体的に特定されていないことから、上記請求項に係る発明は、ソフトウェアによる情報処理がハードウェア資源を用いて具体的に実現されているとはいえない。

したがって、請求項1-4に係る発明は自然法則を利用した技術的思想の創作とは認められないので、「発明」に該当しない(この理由は、例えば各構成要件毎に動作主体を明記すること等により解消される)。

理由2:

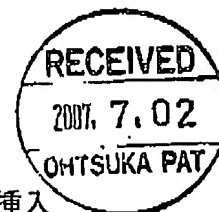
この出願の下記の請求項に係る発明は、その出願前に日本国内又は外国において、頒布された下記 of 刊行物に記載された発明又は電気通信回線を通じて公衆に利用可能となった発明に基いて、その出願前にその発明の属する技術の分野における通常の知識を有する者が容易に発明をすることができたものであるから、特

整理番号:250223

発送番号:322197 発送日:平成19年 7月 2日

2

記 (引用文献等については引用文献等一覧参照)



請求項 1, 5, 6

引用文献 1

引用文献 1 には、文書に対して文書部品の追加挿入の操作を行うと、追加挿入可能な候補として、[前方の節部品、後方に下位節部品、後方に節部品、後方に章部品] が提示されて、選択可能であることが記載されており（特に段落 0004-0009 段落を参照のこと）、上記文書部品として他の文書ファイルを用いることも、当業者ならば容易になし得たものである。

請求項 2

引用文献 1

引用文献 1 には、文書構造をツリー表示することが記載されている。（特に図 36 を参照のこと）

請求項 3

引用文献 1

引用文献 1 においても、注目点によって追加挿入可能な候補が変化することが記載されている。

請求項 4

引用文献 1

追加挿入可能な候補を、どのようなものにするかは、当業者が必要に応じて適宜設定し得た事項にすぎない。

拒絶の理由が新たに発見された場合には拒絶の理由が通知される。

引用文献等一覧

1. 特開平 05-108631 号公報

先行技術文献調査結果の記録

・調査した分野 IPC G06F 17/20-17/26

この先行技術文献調査結果の記録は拒絶理由を構成するものではありません。

<補正等の示唆>

整理番号:250223 発送番号:322197 発送日:平成19年 7月 2日 3/E

と（特許法施行規則様式第13備考6）。

（2）補正は、この出願の出願当初の明細書又は図面に記載した事項のほか、出願当初の明細書又は図面に記載した事項から自明な事項の範囲内で行わなければならない。補正の際には、意見書で、各補正事項について補正が適法なものである理由を、根拠となる出願当初の明細書等の記載箇所を明確に示したうえで主張されたい。（意見書の記載形式は、無効審判における訂正請求書の記載形式を参考にされたい。）

2007. 7. 02

OHTSUKA PAT

この拒絶理由通知の内容に関するお問い合わせ、または面接のご希望がございましたら下記までご連絡下さい。

特許審査第四部電子商取引 浜岸 広明

TEL. 03 (3581) 1101 内線3599

FAX. 03 (3501) 0737

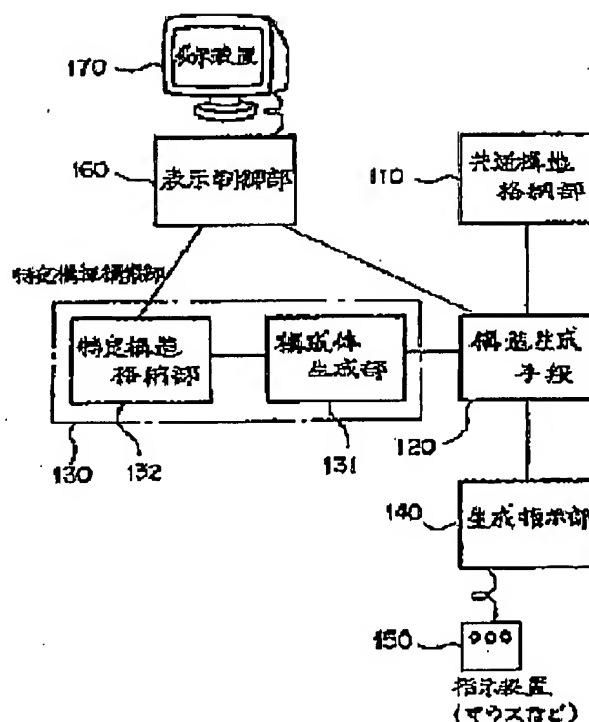
DOCUMENT PROCESSOR

Publication number: JP5108631
Publication date: 1993-04-30
Inventor: SUZUKI KATSUAKI
Applicant: FUJI XEROX CO LTD
Classification:
 - international: G06F17/21; G06F17/21; (IPC1-7): G06F15/20
 - European:
Application number: JP19910267905 19911016
Priority number(s): JP19910267905 19911016

Report a data error here

Abstract of JP5108631

PURPOSE: To select an arbitrary candidate, and also, to generate and edit a user's desired document structure within a range for satisfying a constraint which a common structure has by displaying the whole contents of a candidate of a structure which can be generated at the time of generating and editing a structured document. **CONSTITUTION:** A structure generating means 120 reads out a common structure being logical structure information being common to plural documents, from a common structure store part 110, and also, generates candidates of all specific structures which can be generated, for satisfying a constraint condition for showing a constraint imposed to its common structure, and presents them together with the constraint condition. Its candidate and constraint condition are displayed on a display device 170 through a display control part 160. A generation instructing part 140 informs an instruction request to the structure generating means 120 in order to generate a desired structure body from in a structure body group for constituting each specific structure, from a user. The structure generating means 120 generates and presents a specific logical structure so as to satisfy the constraint condition in accordance with the instruction request from the user inputted through the generation instructing part 140, and also, stores it in a specific structure store part 132.



Data supplied from the esp@cenet database - Worldwide

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平5-108631

(43) 公開日 平成5年(1993)4月30日

(51) Int.Cl. ³	識別記号	序内整理番号	F I	技術表示箇所
G 0 6 F 15/20	5 3 4 P	7343-5L		
	5 3 0 J	7343-5L		

審査請求 未請求 請求項の数 1 (全 46 頁)

(21) 出願番号 特願平3-267905

(22) 出願日 平成3年(1991)10月16日

(71) 出願人 000005496

富士ゼロックス株式会社

東京都港区赤坂三丁目3番5号

(72) 発明者 鈴木 克明

神奈川県横浜市保土ケ谷区神戸町134番地

横浜ビジネスパーク イーストタワー

富士ゼロックス株式会社内

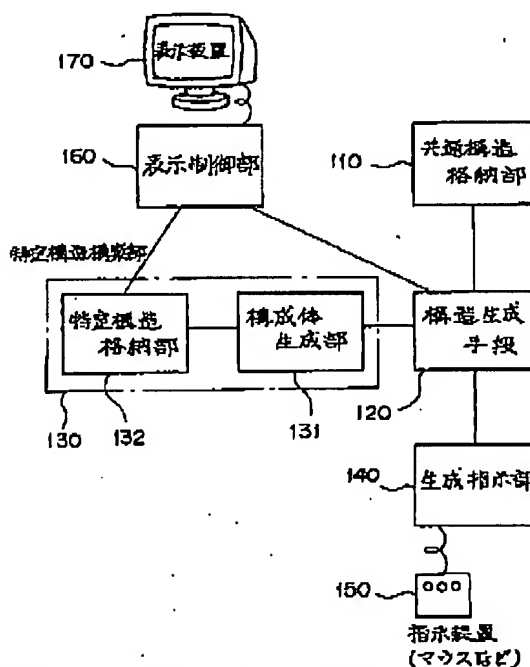
(74) 代理人 弁理士 木村 高久

(54) 【発明の名称】 文書処理装置

(57) 【要約】

【目的】 構造化文書の生成・編集時に生成可能な構造体の候補の全容を表示し、任意の候補の選択を可能とすると共に、共通構造の持つ制約を満足する範囲でユーザの望む文書構造の生成・編集を可能とする文書処理装置を提供すること。

【構成】 構造生成手段120は、共通構造格納部110から、複数の文書に共通な論理構造情報である共通構造を読み出すと共に、その共通構造に課せられている制約を示す制約条件を満足する生成可能な全ての特定構造の候補を生成して、制約条件とともに提示する。その候補及び制約条件は表示制御部160を介して表示装置170に表示される。生成指示部140は、ユーザからの、各特定構造を構成している構成体群の中から所望の構成体を生成すべく指示要求を、構造生成手段120に通知する。構造生成手段120は、生成指示部140を経て入力されたユーザからの指示要求に応じて、上記制約条件を満足するように特定論理構造を生成し提示する。その生成された特定構造は、構成体生成部131によって特定構造格納部132に格納される。



(2)

特開平6-108631

1

2

【特許請求の範囲】

【請求項1】複数の文書に共通な論理構造情報に課せられている制約に基づいて、特定の文書の論理構造情報を、対話的に生成・編集可能な文書処理装置において、前記制約を示す制約条件を満足する生成可能な全ての特定の文書の論理構造情報の候補を生成して、当該制約条件とともに提示するとともに、その提示された候補の中から要求に応じた一つの論理構造情報を生成して提示する構造生成手段を具えたことを特徴とする文書処理装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】この発明は、複数の文書に共通な論理構造情報である共通論理構造による構造上の制約を拘りながら、特定の文書の論理構造情報である特定論理構造を、対話的に生成・編集する機能を有する文書処理装置に関する。

【0002】

【従来の技術】近年、電子化文書が普及し、文書の自動割付けや複数の文書からの抜粋などの文書の自動生成など文書の自動処理の技術が重要になってきている。しかし、大量の文書の自動処理を行うためには自動処理の対象となる文書が、同様な特徴を持っている必要がある。このような目的を達成するためには、例えば複数の文書に共通な情報を持った共通構造と、特定の文書の情報を持った特定構造とによって表現された文書が必要である。そのような文書として、例えば、国際規格であるODA（オープン・ドキュメント・アーキテクチャ：ISO 8613＝Open Document Architecture）における文書がある。この文書は、共通論理構造、共通割付け構造、特定論理構造、特定割付け構造の4つの構造から構成されており、論理構造は章・節・段落といった文書の構成部品の組合せで構成される。

【0003】また従来の構造化文書エディタにおいては、文書構造の作成時に、その時点の制約を評価して生成可能な構成部品の候補の一つを自動的に提供するシステムが提案されている。そのようなシステムでは、文書生成の過程で適応可能な従属子生成規則を順次評価し、その結果、生成可能な構成部品の提示を行ってユーザに判断を仰ぐようにしている。

【0004】例えば構造化エディタの一つである「Griff」（グリフ）（"Interactively editing structured documents" Richard PURITA, Vincent QUINT, Jacques ANDRE Electronic Publishing Vol.1(1)）では編集時に注目している文書の論理構造によって、その前後に追加・挿入できる論理部品が制限される。

【0005】例えば、図36に示すように、「2.1節」、「2.2節」、「2.3節」、「2.4節」の4つの節から構成される第2章において、2.4節（第2章第4節）に注目して部品の追加挿入の操作を行うと、

追加挿入可能な候補として「前方の節部品、後方に下位節部品、後方に節部品、後方に章部品」が提示される。これらの候補は、図36において、例えば前方の節部品として「2.4-」、後方の下位節部品として「2.4.1」、後方の節部品として「2.4+」、後方の章部品として「2+」をそれぞれ追加挿入できることを示している。

【0006】また注目点が2.3節の場合は追加挿入可能な候補として「前方の節部品、後方に下位節部品、後方に節部品」が提示される。これらの候補は、図36において、例えば、前方の節部品として「2.3-」、後方に下位節部品として「2.3.1」、後方に節部品として「2.3+」をそれぞれ追加挿入できることを示している。なお「2.3節」に注目した場合、後方に2.4節が存在するので、章部品の挿入による章（2章）の分割を禁止している。

【0007】このような追加挿入可能な候補の表示例を図37及び図38に示す。

【0008】図37に示す例では、表示画面に表示されたウィンドウ3700に表示されている、「2.1節 CCCCC」と「2.2節 DDDDD」と「2.3節 EEEEE」と「2.4節 FFFFF」の4つの節から構成される第2章（2Second Section）において、「2.4節」に対する部品の追加挿入の操作を行った結果、追加挿入可能な候補の一覧がウィンドウ3710に表示されている。ウィンドウ3710に表示されている候補の中で、「Within selection」は後方に下位節部品を、「Section level 2 before」は前方に節部品を、「Section level 2 after」は後方に節部品を、「Section level 1 after」は後方に章部部品を、それぞれ追加挿入できることを意味している。

【0009】図38に示す例においても、図37に示した第2章において、「2.3節」に対する部品の追加挿入の操作を行った結果、追加挿入可能な候補の一覧がウィンドウ3810に表示されている。

【0010】また上記システム以外の提案されているシステムとして、

「構造化文書における論理論理編集の検討」：山崎英二、鈴鹿千鶴子、杉山孝子、情報処理学会第39回（平成元年後期）全国大会。

の文献に記載されたものが知られている。

【0011】この文献に記載されているシステムでは、構造生成の制約として上記ODAで定義されている従属子生成子を取り入れ、その従属子生成子を評価することにより生成可能な構造を提示するようにしている。

【0012】例えば、「段落部品は下位に文、図、表のいずれか一つの部品を持つ。」という制約があった場合、その制約は、例えば図39に示すような構造（これは共通論理構造に相当する）で表現される。この構造で示される制約により、部品の候補は「文」、「図」、

(3)

特開平5-108631

3

「表」の順に選択される。従ってシステムは最初に文部品を生成してユーザに提示することになる。今、システムが図40(a)に示すような構造を提示し、これに対して、ユーザが「段落3」の文部品を指定して「他の候補」の要求指示を行うと、システムは上記の制約を評価して、図40(b)に示すように、文部品を削除し図部品(候補2)を新たに生成してユーザに提示する。

【0013】このような操作を繰り返すことにより、所望の文書構造(論理構造)を得ることができる。

【0014】

【発明が解決しようとする課題】しかしながら、上記従来の構造化文書エディタでは、所定の文書の制約条件が与えられると、その条件を文書生成の過程で順次評価するので、構成体(部品)の候補を選択して生成が可能となる時点で初めて従属生成規則(制約条件)が評価され生成可能な構成体が提示される。このため、構成体が表示されるまで、どのような構成体が生成可能であるのか分からず、編集開始時には生成可能な文書構造の全体像が全く掴めないのが実情である。

【0015】例えば、以下の制約を持つ文書を作成する場合を考える。

<制約1>文書は一つの序と章の繰り返しの並びからなる。

<制約2>章は段落と、段落か参照の択一選択の反復の任意選択の並びからなる。

<制約3>序は段落の並びからなる。

【0016】この文書は必ず1つの序と1つ以上の章から構成され、章と同じ階層には段落や参照は存在し得ない。序は1つ以上の段落のみから構成され、段落以外の部品を含んではならない。章は必ず1つ以上の段落を最初に含んでいて、その後は段落と参照の繰り返しであり、章の先頭に参照が現れてはならない。

【0017】今、上記の制約を満たす文書として、1つの段落からなる1つの序と、1つの段落からなる1つの章を持った文書が与えられたとする。すなわち、

「序

序第1段落

第1章

1章第1段落」

の文書が与えられたとする。

【0018】「序第1段落」の後ろに同じ階層の構成部品を追加しようとする、制約3が評価され、「段落」のみが生成可能であることが提示される。「1章第1段落」の後ろに同じ階層の構成部品を追加しようとする、制約2が評価され、「段落」あるいは「参照」が生成可能であることが提示される。同様に「1章第1段落」の前に同じ階層の構成部品を追加しようとする、制約2が評価され、「章」の先頭は「段落」でなければならないので、「段落」のみが生成可能であることが提示される。ここで、「1章第1段落」の前に「参照」を

4

追加することができないことは、実際に「1章第1段落」の前に構成部品を追加しようとするまで分からない。

【0019】また、ある階層の構成部品が生成された時点では、更にその下位にどのような構造が生成され得るかは下位の制約条件を評価するまで分からない。「第1章」の後ろに同階層の構成部品を追加しようすると、制約1が評価され、「章」が生成可能であることが提示される。提示に従って「章」を生成すると「第2章」が生成される。

10

【0020】従って、

「序

序第1段落

第1章

1章第1段落

第2章

」

という文書が得られる。

【0021】この時、新たに生成された「第2章」の下位に生成できる構成部品は実際に下位に構成部品を生成しようとして、制約2が評価され「段落」が生成可能であることが提示される。また、1章第1段落の後方に同階層の構成部品を生成しようすると、制約2が評価され、「段落」と「参照」とが生成可能であることが提示される。同様に「序第1段落」の後方に「参照」を生成しようすると、制約3が評価され、「段落」のみ生成可能であることが提示され、「参照」は生成できない。

【0022】従って、

「序

序第1段落

×参照×(生成できない)

第1章

1章第1段落

参照

第2章

2章第1段落

」

という文書が得られる。

【0023】このように上述した従来の構造化文書エディタにおいては、生成できる構成部品(構造)は実際に生成を行うまで不明である。このため、所望の構造の生成が出来なかったときには、上位の選択点まで遡って部分構造(選択点以降の今までに生成された構造)を破棄し、新たな構造(候補)を選択し直さなければならない。また、構造生成のための制約がユーザから見え、制約の評価結果としての生成可能な構造のみが提示されるため、不要な構造を破棄し新たな構造を得るために、どの選択点まで遡れば良いかを知る手立てがなく、試行錯誤を繰り返さなければならない。従って、所望の構造(特定構造)を生成するのに多大な労力を必要としていた。

40

【0024】この発明は、構造化文書の生成・編集時に

(4)

特開平5-108631

5

生成可能な構造体の候補の全容を表示し、任意の候補の選択を可能とすると共に、共通構造の持つ制約を満足する範囲でユーザの望む文書構造の生成・編集を可能とする文書処理装置を提供することを目的とする。

【0025】

【課題を解決するための手段】上記目的を達成するため、この発明では、複数の文書に共通な論理構造情報に課せられている制約に基づいて、特定の文書の論理構造情報を、対話的に生成・編集可能な文書処理装置において、前記制約を示す制約条件を満足する生成可能な全ての特定の文書の論理構造情報の候補を生成して、当該制約条件とともに提示するとともに、その提示された候補の中から要求に応じた一つの論理構造情報を生成して提示する構造生成手段を具えている。

【0026】

【作用】この発明に係る文書処理装置によれば、構造生成手段は、複数の文書に共通な論理構造情報である共通構造に課せられている制約を示す制約条件（例えば従属子生成子）を満足する、生成可能な全ての特定構造の候補を生成して、制約条件とともに提示する。ユーザは、提示された全ての特定構造の候補及び制約条件を参照して、各特定構造の候補を構成している構成体候補群の中から所望の構成体を生成すべく指示要求をする。すると構造生成手段は、ユーザからの指示要求に応じて、制約条件に満足するように特定構造を生成し提示する。

【0027】従って本発明によれば、生成可能な全ての特定構造の候補および制約条件を提示するようにしたので、ユーザは所望する文書の特定構造を容易に作成することができる。

【0028】

【実施例】以下、本発明の実施例を図1乃至図35を参照して説明する。

【0029】図1は本発明に係る文書処理装置の一実施例を機能ブロック図で示したものである。

【0030】同図において、共通構造格納部110は、共通構造（共通論理構造）を格納するものである。

【0031】構造生成手段120は、共通構造格納部110から共通論理構造を読出して、その共通論理構造から、特定構造（特定論理構造）を生成する際の構造上の制約を示す制約条件を読み出すとともに、その制約条件を満足する生成可能な全ての特定の文書の論理構造情報の候補（すなわち論理構造を構成する各要素つまり各構成体になり得る構成体候補群）を生成して、当該制約条件とともに提示する。また提示された候補の中から要求に応じた一つの論理構造情報を生成して提示する。

【0032】特定構造構築部130は、構成体生成部131と特定構造格納部132とから構成されており、生成された構成体に基づいて特定構造を構築する。構成体生成部131は、構造生成手段120によって生成された構成体を特定構造格納部132に格納する。このと

6

き、既に生成されている特定構造に、新たに生成された構成体を接続する。

【0033】生成指示部140は、ユーザがマウス等の指示装置150を操作することにより入力した、構成体を生成すべく要求などの指示情報を、構造生成手段120に通知する。

【0034】表示制御部160は、ディスプレイ等の表示装置170に対して、構造生成手段120によって生成された候補構造や、特定構造格納部132に保持されている処理の対象となっている特定構造（生成過程の構造も含む）等の情報を表示制御する。

【0035】なお、この実施例では、共通構造格納部110及び特定構造格納部132は、ハードディスクやメインメモリなど記憶装置で実現されており、また生成指示部140及び表示制御部160は、プロセッサや中央処理装置等の制御手段で実現されており、更に構造生成手段120及び構成体生成部131は、手段120及び生成部131の各機能を遂行させるためのソフトウェア（プログラム）を上記の制御手段が実行することにより実現される。勿論、ハードウェアやファームウェアで実現することもできる。

【0036】また、この実施例では、上記ODAで規定されている規約に沿って文書の構造を扱っている。上記ODAでは、共通構造から特定構造への制約である従属子生成規則は共通構造を構成する構成体の属性である従属子生成子として格納されている。従属子生成子として格納される従属子生成規則は、その従属子生成子を持つ構成体の直下位に從属することのできる構成体を、その構成体の組み合わせを指定する構造生成式（construction expression）で表現している。その構造生成式を構成する構成規則にはSEQ、REP、OPT、CHO、OPTREPがあるので、次にそれらの規則を説明する。

【0037】◇SEQ（A、B、C…）はA、B、C…の並びを示し、A、B、C…全てが必須であり、出現順番は固定である。

◇REP（A）はAの1回以上任意回の繰り返しを示す。

◇OPT（A）はAの任意選択を示し、Aは出現してもなくても良い。（但し出現は1度でだけである）

◇CHO（A、B、C…）はA、B、C…からの択一選択を示し、A、B、Cのうちどれか一つが出現できる。

◇OPTREP（A）はAの0回以上任意回の繰り返しを示す。

【0038】ここで例えば、

文書は章の繰り返しからなり…制約1

章は段落と、段落が参照の択一選択の反復の任意選択の並びからなる…制約2

という制約は、

<制約1> Doc::=REP（Section）

(5)

特開平5-108631

7

<制約2>Section::=SEQ (Paragraph, OPTREP (CHO (Paragraph, Reference))) (1)
 という構造生成式で表現できる。

【0039】ここで本発明に係る特定構造の作成に際し、構造生成手段120は、前処理として、共通構造から従属子生成規則（構造生成式）を読出し、その規則を満足する最小候補構造を生成して提示すると共に、前記従属子生成規則（構造生成式）を評価することにより最小構造を生成して提示する。その生成された最小候補構造及び最小構造の情報は、表示制御部160を経て表示装置170に入力され表示される。なお最小候補構造及び最小構造の生成の詳細については後述する。

【0040】この実施例では、特定構造の作成・編集操作の際には、表示装置170の表示画面には、図2に示すように、候補構造を表示するための候補構造表示領域210と、候補構造表示領域210に表示されている候補構造に対応する特定構造を表示するための特定構造表示領域220とを有するウィンドウ200が表示されるようになっている。

【0041】図2において、候補構造表示領域210には、上記式(1)の構造生成式を満足する最小候補構造210Aが表示されており、特定構造表示領域220には、上記式(1)の構造生成式を評価して得られた最小構造220Aが表示されている。因みに最小構造220Aは、最小候補構造210A中の「REP」及び「SEQ」の構成規則が評価されて得られたものである。

【0042】なお、この実施例では、最小候補構造を含む候補構造（最小候補構造に新たな構造が追加された構造）を構成する各要素をノードと定義し、これらのノードのうち例えば最小候補構造210Aを構成する各ノードで示されるように、「Logical Root Class」などルート（根）に相当するノードをルートノードと定義し、また、「Section」、2つの「Paragraph」、「Reference」など構成体になり得る候補を表しているノードを候補ノードと定義し、更に、「REP」、「SEQ」、「OPTREP」、「CHO」など構成規則を表しているノードをGFS (Generator for subordinates) ノードと定義する。

【0043】ところで、図2に示す最小候補構造210Aに対して、構造の追加、たとえば「REP」や「OPTREP」のGFSノードを複数回適用（複数回評価）することにより、そのノード以下の構造を新たに追加できるので、候補構造表示領域210には常に最小候補構造が表示されるとは限らない。同様に、例えば最小候補構造210A中のGFSノード「OPTREP」が評価されることにより、そのノードの下位に存在している候補ノード「Paragraph」に対応する構成体が生成され、その構成体が最小構造220A中の構成体「aSection」の下位に接続されるので、特定構造表示領域220には常に最小構造が表示されるとは限らない。

8

【0044】またこの実施例では、あるGFSノードが下位ノードとして持っている構造を候補部分構造と定義する。例えば最小候補構造210Aにおいて、GFSノード「SEQ」が下位ノードとして持っている、図2中符号230で示される点線内の構造が候補部分構造であり、同様に、GFSノード「OPTREP」が下位ノードとして持っている、図2中符号240で示される点線内の構造が候補部分構造である。

【0045】更に、候補部分構造に対応して生成されたものを特定構造における部分構造と定義する。例えば候補部分構造230において、GFSノード「OPTREP」が選択されていなければ、候補ノード「Paragraph」に対応する構成体が部分構造となり、同様に候補部分構造240において、GFSノード「CHO」の規則により候補ノード「Paragraph」が選択されているのであれば、候補ノード「Paragraph」に対応する構成体が部分構造となる。また候補部分構造230において、例えばGFSノード「OPTREP」が選択され、かつ、GFSノード「CHO」の規則により候補ノード「Reference」が選択されているのであれば、候補ノード「Paragraph」及び「Reference」に対応する各構成体からなる構造が部分構造となる。

【0046】次に、構造生成手段120による特定構造生成・編集処理について、図3乃至図8を参照して説明する。

【0047】なお、図3は特定構造生成・編集処理の全体の処理動作を示すフローチャートを、図4は最小候補構造の生成処理動作を示すフローチャートを、図5は従属子生成子の展開処理動作を示すフローチャートを、図6は構造生成式の展開処理動作を示すフローチャートを、図7は最小構造の生成処理動作を示すフローチャートを、図8は下位構造の生成処理動作を示すフローチャートを、それぞれ示している。

【0048】図3に示すように、構造生成手段120は、共通構造格納部110から共通構造を読出すとともに（ステップ310）、最小候補構造生成処理用のサブルーチン呼び出して、読出した共通構造の制約から最小候補構造の生成を行う（ステップ320）。その後、表示制御部160に対して、その最小候補構造を送出して表示装置170に表示させる（ステップ330）。次に、最小構造生成処理用のサブルーチン呼び出して、生成された最小候補構造から最小構造の生成を行うとともに（ステップ340）、表示制御部160に対して、その最小構造を送出して表示装置170に表示させる（ステップ350）。続いて、編集操作が終了した否かを判断し（ステップ360）、終了した場合には処理を終了し、一方終了していない場合は、編集指示要求処理用のサブルーチンを読出して、ユーザからの編集操作に対応した生成指示部140からの指示に従って構造（候補部分構造および部分構造）の生成を行うとともに（ス

(6)

特開平5-108631

9

10

ステップ370)、その構造の生成に応じて、候補構造及び特定構造の表示を更新する(ステップ380)。ステップ380終了後は上記ステップ360に戻りこのステップ以降を実行する。

【0049】次に最小候補構造生成処理について図4を参照して説明する。

【0050】最小候補構造生成処理用のサブルーチンが呼び出されると、構造生成手段120は、図4に示すように、文書の論理根(Root)の属性GFS(Generator for subordinates)としての従属子生成子を読み込むとともに(ステップ410)、属性GFS展開処理用のサブルーチンを呼び出して、属性GFSの展開処理を行う(ステップ420)。ステップ420を終了したら、図3のステップ320にリターンする。

【0051】次に属性GFS展開処理について図5を参照して説明する。

【0052】属性GFS展開処理用のサブルーチンが呼び出されると、構造生成手段120は、図5に示すように、「GFS=nll」(GFSが定義されていない)の式が成立するかどうかを判断する(ステップ510)。ここで、属性GFSが定義されている場合(上記「GFS=nll」の式が不成立)は、構造生成式展開処理用のサブルーチンを呼び出して、GFSの構造生成式(construction expression:CEと略記)を展開する(ステップ520)。ステップ520を終了した場合、上記ステップ510で属性GFSが定義されていない場合は、図4のステップ420にリターンする。

【0053】次に属性GFSの構造生成式展開処理について図6を参照して説明する。

【0054】構造生成式展開処理用のサブルーチンが呼び出されると、構造生成手段120は、図6に示すように、構造生成式の構成規則に対応するGFSノード(GFS-nodo)を生成して、上位ノード(上位nodo)の子供として繋ぐとともに(ステップ610)、構造生成式の引数を読み出して(ステップ620)、先頭の引数を取り出す(ステップ630)。

【0055】次に、引数=CE(構造生成式)の式が成立するかどうかを判断する(ステップ640)。

【0056】ここで、上記式が不成立の場合つまり引数が構成体の場合は、構成体に対応する候補ノード(候補-nodo)を生成して、GFSノードの子供として繋ぐとともに(ステップ650)、属性GFS展開処理用のサブルーチン(図5参照)を再帰的に呼び出して、その構成体の属性GFSを展開する(ステップ660)。

【0057】一方、ステップ640で「引数=CE」の式が成立する場合つまり引数が構造生成式の場合は、構造生成式展開処理用のサブルーチンを再帰的に呼び出して、生成したGFSノードを上位ノードとして構造生成式を展開する(ステップ670)。

【0058】上記ステップ660或いはステップ670

を終了したら、構造生成手段120は、上記ステップ620において読出した構造生成式の全ての引数を評価したか否かを判断し(ステップ680)、未評価の引数がある場合には、次の引数を取り出し(ステップ690)、その後上記ステップ640に戻りこのステップ以降を実行し、一方、全て評価した場合は図5のステップ520にリターンする。

【0059】次に最小構造生成処理について図7を参照して説明する。

【0060】最小構造生成処理用のサブルーチンが呼び出されると、構造生成手段120は、図7に示すように、ルートノード(Rootノード)に対応する構成体を生成し(ステップ710)、その構成体に下位ノードが存在するか否かを判断する(ステップ720)。

【0061】ここで、下位ノードが存在している場合は、下位構造生成処理用のサブルーチンを呼び出して、その後、上記ステップ710において生成した構成体から下位ノードを取り出して、下位構造の生成を行う(ステップ730)。そして、ステップ730を終了した場合、上記ステップ720で下位ノードが存在しない場合は、図3のステップ340にリターンする。

【0062】次に下位構造生成処理について図8を参照して説明する。

【0063】下位構造生成処理用のサブルーチンが呼び出されると、構造生成手段120は、図8に示すように、GFSノードであるか否かを判断し(ステップ801)、判断結果に応じた処理(ノードの評価)を行う。

【0064】ここで、候補ノードである場合は対応する構成体を生成して、該生成した構成体を、最近、上位の候補ノードで生成された構成体の子供として繋ぐ。

【0065】これに対し、GFSノードである場合はGFSの構造生成式に応じて以下のような処理を行う。

【0066】◇SEQ:直属下位のノードを先頭から順に全て評価する。

◇REP:直属下位のノードを1度評価する。

◇OPT:評価を打切る。

◇CHO:直属下位のノードの先頭だけを評価する。

◇OPTREP:評価を打切る。

【0067】そこで、上記ステップ801において、GFSノードの場合は、「CE=SEQ」であるか否かを判断し(ステップ802)、ここでSEQの場合は、下位ノードの先頭を取り出し(ステップ803)、その後、下位構造生成処理用のサブルーチンを呼び出して下位構造の生成を行う(ステップ804)。次に、全ての下位ノードを処理したか否かを判断し(ステップ805)、未処理の下位ノードがある場合は次の下位ノードを取り出して(ステップ806)、上記ステップ804に戻りこのステップ以降を実行する。

【0068】ステップ802で「CE≠SEQ」の場合は「CE=REP」であるか否かを判断し(ステップ8

(7)

特開平5-108631

11

07)、ここでREPの場合は、下位ノードを取り出し(ステップ808)、その後、下位構造生成処理用のサブルーチン呼び出して下位構造の生成を行う(ステップ809)。

[0069] ステップ807で「CE≠REP」の場合は「CE=CHO」であるか否かを判断し(ステップ810)、ここでCHOの場合は、下位ノードの先頭を取り出し(ステップ811)、その後、下位構造生成処理用のサブルーチン呼び出して下位構造の生成を行う(ステップ812)。

[0070] ステップ810で「CE≠CHO」の場合は「CE=OPT」か否かを判断し(ステップ813)、ここでOPTの場合も何もしない。

[0071] 上記ステップ813で「CE≠OPT」の場合は「CE=OPTREP」か否かを判断し(ステップ814)、ここでOPTREPの場合は何もしない。

[0072] 上記ステップ801においてGFSノードでない場合つまり候補ノードの場合は、候補ノードに対応する構成体を生成し、その構成体を最近の上位の候補ノードから生成された構成体の子供として繋ぐ(ステップ815)。次に、その構成体の下位ノードが存在するか否かを判断し(ステップ816)、ここで下位ノードが存在する場合は、下位ノードを取り出し(ステップ817)、その後、下位構造生成処理用のサブルーチン呼び出して下位構造の生成を行う(ステップ818)。

[0073] なお、上記ステップ805において全ての下位ノードを処理した場合、上記ステップ809、ステップ812、ステップ818のいずれかを終了した場合、上記ステップ813あるいはステップ814で「YES」の場合、上記ステップ816において下位ノードが存在しない場合、のいずれかの場合は、図7のステップ730にリターンする。

[0074] なお候補構造を保持するデータ構造については、本願出願人が先に出願した特願平2-157067号に開示されているデータ構造(ジェネレータ)を用いることにより実現できる。また最小構造の生成は、本願出願人が先に出願した特願平2-157068号に開示されている、割付け可能木の最小構造生成の方法を用いることにより実現できる。

[0075] 以上の処理で最小候補構造及び最小構造が生成されたことになる。

[0076] そして、ユーザが指示装置150を操作してGFSノードの選択を行うことにより、その選択情報が生成指示部141を経て構造生成手段120に入力されると、生成手段120は、入力した選択情報に基づくGFSノードの構成規則に応じた構造生成処理の候補を、表示制御部160に送出する。すると表示制御部160は、入力した構造生成処理の候補を表示装置170に表示する。

[0077] ここで、この実施例において定義した属性

12

GFSの構成規則に応じた適応可能な構造生成処理について構成規則毎に説明する。

[0078] ◇SEQでは無し

◇REPでは処理[add subordinate]と処理[delete subordinate]とがある。[add subordinate]は従属するノードの追加。下位ノードとして持っている候補部分構造(選択されているノードを根として展開した候補構造)を末っ子として、既に生成されている候補構造に追加し、更に候補部分構造に対応する部分構造(選択されているノードを根として生成した構造)を生成して、既に生成されている特定構造に追加する。[delete subordinate]は従属するノードの削除。下位ノードとして2つ以上のノードが生成されている場合、最後のノードを候補構造から削除し、そのノードに対応する構成体を、既に生成されている特定構造から削除する。

[0079] ◇OPTでは処理[select]と処理[reject]とがある。[select]は下位のノードを選択する。下位のノードとして持っている候補部分構造に対応する部分構造を生成して、既に生成されている特定構造に追加する。

[reject]…下位のノードの非選択。

下位のノードとして持っている候補部分構造に対応する部分構造を、既に生成されている候補構造から削除する。

[0080] ◇CHOでは処理[n-subordinate]とがある。[n-subordinate]はn番目の下位ノードを選択する。以前に選択されていた下位のノードとして持っている候補部分構造に対応する部分構造を、既に生成されている特定構造から削除し、新たに選択されたノードに対応する部分構造を生成して、その特定構造に追加する。

[0081] ◇OPTREPでは処理[select]と処理[reject]と処理[add subordinate]と処理[delete subordinate]とがある。[select]、[reject]はOPTの場合と同様。[add subordinate]は従属するノードの追加。これはREPの場合と同様であるが、下位ノードが選択されているときのみ作用する。[delete subordinate]は従属するノードの削除。これはREPの場合と同様であるが、下位ノードが選択されているときのみ作用する。

[0082] 次に、上述した最小候補構造に対する編集指示要求の処理を図9を参照して説明する。

[0083] 図9に示すように、ユーザからの編集要求が発せられたGFSノードを求め、そのGFSノードの持つ構成規則を調べ(ステップ901)、その結果に応じた処理を行う。

[0084] 上記ステップ901において「CE=REP」であるか否かを調べ(ステップ902)、REPの場合は、ここで適用可能な編集処理の一覧

[add subordinate]、[delete subordinate]

(8)

特開平5-108631

13

をユーザに提示し(ステップ903)、更にGFSノード[REP]における編集操作処理用のサブルーチンを呼び出して、ユーザの指示に従って構成体の生成を行う(ステップ904)。

【0085】上記ステップ902においてREPでない場合は、「CE=CHO」であるか否かを調べ(ステップ905)、CHOの場合は、ここで適用可能な編集一覧

[1- subordinate]、…、[n- subordinate]

をユーザに提示し(ステップ906)、更にGFSノード[CHO]における編集操作処理用のサブルーチンを呼び出して、ユーザの指示に従って構成体の生成を行う(ステップ907)。

【0086】上記ステップ905においてCHOでない場合は、「CE=OPT」であるか否かを調べ(ステップ908)、OPTの場合は、ここで適用可能な編集一覧

[select]、[reject]

をユーザに提示し(ステップ909)、更にGFSノード[OPT]における編集操作処理用のサブルーチンを呼び出して、ユーザの指示に従って構成体の生成を行う(ステップ910)。

【0087】上記ステップ908においてOPTでない場合は、「CE=OPTREP」であるか否かを調べ(ステップ911)、OPTREPの場合は、ここで適用可能な編集一覧

[select] [reject] [add subordinate] [delete subordinate]

をユーザに提示し(ステップ912)、更にGFSノード[OPTREP]における編集操作処理用のサブルーチンを呼び出して、ユーザの指示に従って構成体の生成を行う(ステップ913)。

【0088】なお、上記ステップ904、ステップ907、ステップ910、ステップ913のいずれかのステップを終了した場合は、図3のステップ370にリターンする。

【0089】次に、構造生成手段120による、呼び出された編集操作処理用のサブルーチンの実行について、各GFSノード毎に説明する。

【0090】GFSノード[REP]における編集操作のアルゴリズム。図10に示すように、適用された編集処理を調べる(ステップ1001)。処理が[add subordinate]であるかを調べ(ステップ1002)、そうであれば、下位ノードとして持っている候補部分構造を末っ子として、既に生成されている候補構造に追加し、更に候補部分構造に対応する部分構造を生成して、既に生成されている特定構造に追加する(ステップ1003)。上記ステップ1002で[add subordinate]でなければ、処理が[delete subordinate]であるかを調べ(ステップ1004)、そうであれば、下位ノードが

14

2つ以上あるか否かを調べる(ステップ1005)。ここで2つ以上あれば、下位ノードの最後のノードを候補構造から削除し、その削除されたノードに対応する構成体を特定構造から削除する(ステップ1006)。なお、上記ステップ1003あるいはステップ1006を終了した場合、上記ステップ1005において下位ノードが1つの場合は、図9のステップ904にリターンする。

【0091】GFSノード[CHO]における編集操作のアルゴリズム。図11に示すように、適用された編集処理を調べ(ステップ1101)、以前に選択されていた下位のノードとして持っている候補部分構造に対応する部分構造を、既に生成されている特定構造から削除し(ステップ1102)、その後、選択された下位ノードとして持っている候補部分構造に対応する部分構造を生成して、ステップ1102における特定構造に追加する(ステップ1103)。ステップ1103終了後は、図9のステップ907にリターンする。

【0092】GFSノード[OPT]における編集操作のアルゴリズム。図12に示すように、適用された編集処理を調べる(ステップ1201)。処理が[select]であるかを調べ(ステップ1202)、そうであれば、下位ノードとして持っている候補部分構造に対応する部分構造を生成して、既に生成されている候補構造に追加する(ステップ1203)。上記ステップ1202において[select]でなければ、処理が[reject]であるかを調べ(ステップ1204)、そうであれば、下位ノードとして持っている候補部分構造に対応する部分構造を、既に生成されている候補構造から削除する(ステップ1205)。なお、上記ステップ1203あるいはステップ1205を終了したら、図9のステップ910にリターンする。

【0093】GFSノード[OPTREP]における編集操作のアルゴリズム。図13に示すように、適用された編集処理を調べる(ステップ1301)。処理が[select]であるかを調べ(ステップ1302)、そうであれば、下位のノードとして持っている候補部分構造に対応する部分構造を生成して、既に生成されている特定構造に追加する(ステップ1303)。上記ステップ1302で[select]でなければ、処理が[reject]であるかを調べ(ステップ1304)、そうであれば、下位のノードとして持っている候補部分構造に対応する部分構造を、既に生成されている特定構造から削除する(ステップ1305)。

【0094】上記ステップ1304で[reject]でなければ、処理が[add subordinate]であるかを調べ(ステップ1306)、そうであれば、ノードが選択(select)されているか否かを判断し(ステップ1307)、ここで選択(select)されていれば、下位ノードとして持っている候補部分構造を末っ子として、既に生成され

(9)

特開平5-108631

15

ている候補構造に追加し、更に候補部分構造に対応する部分構造を生成して、既に生成されている特定構造に追加する(ステップ1308)。

【0095】上記ステップ1306で[add subordinate]でなければ、処理が[delete subordinate]であるかを調べ(ステップ1309)、そうであれば、ノードが選択(select)されているか否かを判断し(ステップ1310)、ここで選択(select)されていれば、下位ノードが2つ以上あるか否かを判断する(ステップ1311)。ここで下位ノードが2つ以上ある場合は、下位ノードの最後のノードを候補構造から削除し、その削除されたノードに対応する構成体を特定構造から削除する(ステップ1312)。

【0096】なお、上記ステップ1303、ステップ1305、ステップ1308、ステップ1312のいずれかのステップを終了した場合、上記ステップ1307或いはステップ1310においてノードが選択されていない場合、上記ステップ1311において下位ノードが1つしかない場合は、図9のステップ913にリターンする。

【0097】次に、最小候補構造、最小構造、及び特定構造の生成処理を、具体例を挙げて説明する。

【0098】制約として、

<制約1> Doc::=REP (Section)

<制約2> Section::=REP (SEQ (Paragraph, OPTREP (CHO (Paragraph, SEQ (Reference No, Referred No, Reference Body))))))

なる構造生成式 (construction expression : CEと略記) が与えられた場合は、上述した図4、図5及び図6のフローチャートに基づく処理を行うことにより、最小候補構造が生成されることになる。

【0099】すなわち、

(1) 制約1である属性GFSのDOC::REP (Section) が展開される。

【0100】(2) GFSノード[REP]が生成され根であるDocの直下に繋がれ、更に引数[Section]が評価される。

【0101】(3) [Section]は構成体であるので候補ノード[Section]が生成されGFSノード[REP]の子供として繋がれる。

【0102】(4) 次に[Section]の属性GFSの制約2が展開される。

【0103】(5) GFSノード[REP]が生成され候補ノード[Section]の子供として繋がれる。

【0104】(6) 構成規則REPの引数が評価され、GFSノード[SEQ]が生成されGFSノード[REP]の子供として繋がれる。

【0105】(7) 構成規則SEQの引数が順に評価される。

【0106】すなわち構成規則SEQの第1引数[Para

16

graph]が評価され候補ノード[Paragraph]がGFSノード[SEQ]の子供として繋がれ、更に[Paragraph]のGFSが展開される。その[Paragraph]は属性GFSを持たないので評価がとまる。

【0107】そして構成規則SEQの第2引数のCE[OPTREP (CHO (Paragraph, SEQ (Reference No, Referred No, Reference Body)))]の展開を行う。

【0108】同様に、構成規則CHOの第1引数[Paragraph]、更に構成規則SEQの第1引数[Reference No]、第2引数[Referred No]、第3引数[ReferenceBody]をそれぞれ評価すると、その結果として図14に示されるウィンドウ1400の候補構造表示領域1400Aに表示されているように、候補構造1410が生成されることとなる。この候補構造1410が最小候補構造であり、この構造に対して構造の追加、削除が行われる。特定構造表示領域1300Bにはオブジェクト(Object、以下Objという)1 (aLogical RootClass)が生成されている。なお、表示領域1400Bに表示されているOBJ1 (「Logical Root Class」)が、上述した構造生成式(制約1)における「Doc」に対応するものである。

【0109】最小候補構造が生成されると、図7、図8のフローチャートに基づく処理により、この最小候補構造を元に最小構造が生成されることとなる。

【0110】すなわち、最小候補構造からは、ルート(Root)ノードからルートの構成体が生成される。ルートノードの下位ノードは「REP」のGFSノードなので更に下位のノードが評価され「Section」の構成体が生成されルートの構成体の子供として繋がれる。

「Section」の候補ノードの下位ノードは「REP」であるから、更に下位の「SEQ」のGFSノードが評価され「Body Paragraph」の構成体が「Section」の構成体に繋がれる。そして「OPTREP」のGFSノードで評価が打ち切られる。この結果として図15に示されるウィンドウ1400の表示領域1400Bに表示されているように、特定構造1500が生成されることとなる。

【0111】特定構造1500は、Obj1 (aLogical Root Class)の下位としてObj2 (aSection)が生成され、更にObj2の下位としてObj3 (aBody Paragraph)が生成されている。

【0112】以上の処理により最小候補構造から最小構造が生成されたことになる。

【0113】次に、最小候補構造と最小構造とに基づいて、制約を満たした特定論理構造の編集例を、図16乃至図26を参照して説明する。

【0114】ユーザが、図15に示す表示状態から、指示装置によりGFSノードの選択を行い、構造生成手段が起動されると、構造生成手段120は選択されている

(10)

特開平5-108631

17

GFSノードの構成規則に応じた構造生成処理の候補をユーザに提示する。

【0115】ユーザが、例えば図16に示すように、ウィンドウ1400の表示領域上で指示装置150によりGFSノード1600（GFSノード【OPTREP】）を選び、マウスボタンのクリックなどにより編集操作要求を出すと、図9のアルゴリズムにより生成規則【OPTREP】に対応する適用可能処理（構造生成処理の候補）の一覧が表示される。その表示状態を図17に示す。図17において、1700は適用可能処理の一覧であり例えばメニュー形式の一覧である。その適用可能処理の一覧内の所望の項目を選択することにより、構造生成手段120によって該当する処理が行われる。

【0116】今、ユーザが、図17の表示状態から図18に示すように適用可能処理の一覧1700における【Select】を指示すると、GFSノード【OPTREP】の下位ノードに対して最小構造生成処理が適用され、新たな構成体（aBody Paragraph）が生成される。この時点での特定構造の表示状態を図19に示す。

【0117】図19において、1900は特定構造であり、この特定構造1900は、図15に示した特定構造1500の構成においてObj2の下位としてObj4（aBody Paragraph）が生成され追加された構成になっている。

【0118】次にユーザが図20に示すようにGFSノード2000（GFSノード【CHO】）を選択することにより、生成規則【CHO】に対応する適用可能処理の一覧が表示されるので、ユーザはその一覧の中から所望の処理を選択する。図21は、生成規則【CHO】に対応する適用可能処理の一覧2100が表示され、その一覧の中から【2-subordinate】（2番目の選択肢）が指示されたことを示している。

【0119】このようにして生成規則【CHO】が選択され、処理【2-subordinate】が選択されると、図22に示す表示領域1400Bに表示されているような、特定構造2200が得られる。この特定構造2200は、図19に示した特定構造1900の構成において、Obj2の下位としてのObj4（aBody Paragraph）が破棄され、そしてObj2の下位として、新たに選択されたGFSノード2210（GFSノード【SEQ】）以下の構成体の候補に対応して、Obj5（aReference No）、Obj6（aRefered No）、及びObj7（aReferenceBody）がそれぞれ生成され接続された構成になっている。

【0120】同様にして、ユーザが、図22に示す表示状態から、図23に示すように例えばGFSノード2300（GFSノード【REP】）を選択し、更に編集操作要求を行うとともに、図24に示すように、その結果として表示されたGFSノード【REP】の適用可能処理の一覧2400の中から【add subordinate】を指示

18

すると、図23に示すGFSノード2300（GFSノード【REP】）の下位として、そのGFSノード2300以下の候補部分構造（図23中矢印Aで示される点線内の構造）が生成される。この時点での、候補部分構造及び特定構造の表示状態を図25に示す。

【0121】図25において、表示領域1400A内の候補構造2500AにおけるGFSノード2300（GFSノード【REP】）の下位として、図23中矢印Aで示される点線内の構造が生成された構造2510が繋がっている。

【0122】一方、表示領域1400Bには候補構造2500に対応する特定構造2500Bが表示されている。この特定構造2500Bは、図22に示した特定構造2200の構成に、候補部分構造2510が追加生成されたことによりGFSノード2520（GFSノード【SEQ】）が評価され、その結果として構成体（aBody Paragraph）が生成されて、その構成体がObj8としてObj2の下位に繋がった構成になっている。

【0123】そして、特定構造2500Bが所望の構造であったとすれば、特定構造2500Bが最終的な特定構造つまり特定論理構造となる。

【0124】同様な操作を繰り返すことにより、多種多様な特定構造を生成できる。その特定構造の一例を図26に示す。

【0125】図26に示す候補構造2600Aは、図25の表示領域1400Aに表示されている候補構造において、GFSノード2300（GFSノード【REP】）に対する【add subordinate】の処理操作を1回行い、次にその結果得られた候補構造において、図25に示すGFSノード2530（GFSノード【REP】）に相当するGFSノード【REP】に対する【add subordinate】の処理操作を1回行い、更にその結果得られた候補構造において、図25に示すGFSノード2540（GFSノード【OPTREP】）に相当するGFSノード【OPTREP】に対する【add subordinate】の処理操作を2回行うことにより得られることになる。こうして得られた候補構造2600Aにおいて、候補ノード2601～2613が生成されることにより、特定構造2600Bが得られることになる。

【0126】上述したようにこの実施例によれば、生成可能な特定構造の候補の全容を提示しながら、共通構造から制約を満たしつつ特定構造の編集が可能である。

【0127】更にこの実施例では、GFSノードはその下位にいくつのノードを持っているかといった状態を保持することができるので、その状態の情報を活用することによって構造編集の簡単なガイダンスを行うことができる。

【0128】例えば、GFSノード【REP】での適用可能処理（な編集処理）【add subordinate】、【delete subordinate】であるが、【delete subordinate】は

(11)

特開平5-108631

19

下位ノードが2個以上のときのみ有効である。そこで、GFSノードで編集操作要求があった時に、そのGFSノードの状態を調べ、有効な編集操作のみをユーザに提示する編集装置が実現可能である。

【0129】ここに、GFSノードの各ノードについて説明する。

【0130】◇REP

(a) 下位ノードが2個未満のとき、処理 [add subordinate] のみ有効。

(b) 下位ノードが2個以上のとき、処理 [add subordinate] 及び処理 [delete subordinate] 共に有効。

【0131】◇OPT

(a) 下位ノードが選択 (select) されているとき、処理 [disselect] のみ有効。

(b) 下位ノードが選択されていないとき、処理 [reject] のみ有効。

【0132】◇OPTREP

(a) 下位ノードが選択されていないとき、処理 [select] のみ有効。

(b) 下位ノードが選択されていて、かつ、下位ノードが2個未満のとき、処理 [reject]、処理 [add subordinate] が有効。

(c) 下位ノードがselectされていて、かつ、下位ノードが2個以上のとき、処理 [reject]、処理 [add subordinate]、処理 [delete subordinate] の各処理が有効。

【0133】次に、GFSノードの状態によって有効な適用可能処理 (編集処理) が制限される様子を、図27乃至図35に示す。

【0134】最小候補構造及び最小構造を図27に示す。

【0135】図27に示すように、ウィンドウ2700の候補構造表示領域2700Aには最小候補構造2710が表示され、また特定構造表示領域2700Bには最小構造2720が表示されている。

【0136】なお最小候補構造2710において、「Doc」、「AAA」、「BBB」、「CCC」、「DDD」、「EEE」、「FFF」は候補ノードであり、「REP」、2つの「SEQ」、「OPT」、「OPTREP」、「CHO」はGFSノードである。一方最小構造2620において、「aDoc」は論理根であり、「aAAA」、「aBBB」は構成体である。

【0137】ユーザが、図27に示す表示状態から、図28に示すようにGFSノードの1つである「OPT」を選択し、更に編集処理要求を行うと、図29に示す適用可能処理の一覧2900が表示される。これは、GFSノード「OPT」の下位ノードが選択されていないので、有効な処理として「select」が表示されたのである。

【0138】次にユーザが、図30に示すように、一覧

20

2900内の「select」を指示すると、GFSノード「OPT」が評価される。その結果として図31に示される表示領域2700Bに表示されている特定構造3100が得られる。

【0139】特定構造3100の生成に際し、GFSノード「OPT」が処理「select」に従って評価されるので、部分構造として、構成体「aCCC」とその下位として構成体「aDDD」が生成され、更にその部分構造が、構成体「aAAA」に繋がることにより、特定構造3100が生成されることとなる。

【0140】ユーザが、図31に示す表示状態から、再度、GFSノード「OPT」を選択し、編集処理要求を行うと、GFSノード「OPT」の下位ノードとして「CCC」、「DDD」の各候補ノードが選択されているので、今度は、図31に示すように、有効な処理として「reject」のみが表示されている適用可能処理の一覧3200が表示される。

【0141】またユーザが、図31に示す表示状態から、GFSノード「OPTREP」を選択し、編集処理要求を行うと、そのGFSノード「OPTREP」の下位ノードが選択されていないので、有効な処理として「select」のみが表示されている適用可能処理の一覧が表示される。そして図33に示すように、その適用可能処理の一覧3300内の「select」を指示すると、GFSノード「OPTREP」が評価され、その結果として、図34に示す表示領域2700Bに表示されている特定構造3400が得られることとなる。

【0142】なお、図34は、GFSノード「OPTREP」に対する1回目の編集処理操作が行われて部分構造として構成体「aEEE」が生成され、更に2回目の編集操作が行われて適用可能処理の一覧3410が表示された状態を示したものである。

【0143】適用可能処理の一覧3410には、下位ノードとして候補ノード「EEE」が選択され、かつ、下位ノードが2個未満なので、「add subordinate」及び「reject」の処理が表示されている。この場合は、GFSノード「OPTREP」には4つの編集処理が設けられているにも関わらず、上述した規則により、編集処理が2つに制限されたのである。

【0144】次にユーザが、図34に示す表示状態から、適用可能処理の一覧3410の中から、「add subordinate」を指示すると、下位ノードとして持っている候補部分構造 (つまりCHO (「EEE」「FFF」)) を末っ子として、既に生成されている特定構造に追加するとともに、候補部分構造に対応する部分構造 (つまりこの場合は「aEEE」) が生成される。

【0145】同様にして、GFSノード「OPTREP」に対する、「addsubordinate」の編集処理操作を3回繰り返すと、結果として、図35に示す表示領域2700Bに表示されている特定構造3500が得られる

(12)

特開平5-108631

21

こととなる。なお図35の表示領域2700Bにおいて、矢印Aで示される構成体「aEEE」はGFSノード「OPTREP」に対する「select」の処理操作（1回目の操作）によって生成されたものであり、また矢印Bで示される構成体「aEEE」、矢印Cで示される構成体「aEEE」はそれぞれGFSノード「OPTREP」に対する「addsubordinate」の処理操作（2回目、3回目の操作）によって生成されたものである。

【0146】なお、図35は、GFSノード「OPTREP」に対する3回目の編集処理操作が行われて部分構造として矢印Aで示される構成体「aEEE」が生成され、更に4回目の編集操作が行われて編集処理の一覧3510が表示された状態を示したものである。

10

【0147】

【発明の効果】以上説明したように本発明によれば、構造生成手段は、複数の文書に共通な論理構造情報に課せられている制約を示す制約条件を満足する生成可能な全ての特定の文書の論理構造情報の候補を生成して、当該制約条件とともに提示するとともに、その提示された候補の中から要求に応じた一つの論理構造情報を生成して提示するようにしたため、ユーザは、構造化文書の生成・編集時に、生成可能な構造体の候補の全容を知ることができるとともに、任意の候補の選択をして、所望の文書構造の生成・編集を行うことができる。

20

【図面の簡単な説明】

【図1】本発明に係る文書処理装置の一実施例を示す機能ブロック図。

【図2】本発明に係る特定構造の生成を説明するための図。

【図3】本実施例の特定構造生成・編集の全体の処理動作を示すフローチャート。

【図4】本実施例の最小候補構造の生成処理動作を示すフローチャート。

【図5】本実施例の従属子生成子の展開処理動作を示すフローチャート。

【図6】本実施例の構造生成式の展開処理動作を示すフローチャート。

【図7】本実施例の最小構造の生成処理動作を示すフローチャート。

【図8】本実施例の下位構造の生成処理動作を示すフローチャート。

【図9】本実施例の編集指示要求の処理動作を示すフローチャート。

【図10】本実施例の従属子生成子「REP」における編集操作の処理動作を示すフローチャート。

【図11】本実施例の従属子生成子「CHO」における編集操作の処理動作を示すフローチャート。

【図12】本実施例の従属子生成子「OPT」における編集操作の処理動作を示すフローチャート。

50

【図13】本実施例の従属子生成子「OPTREP」に

22

における編集操作の処理動作を示すフローチャート。

【図14】本実施例における最小候補構造を説明するための図。

【図15】本実施例における最小構造を説明するための図。

【図16】本実施例における最小候補構造から特定構造を得るための生成過程を説明するための図。

【図17】本実施例における最小候補構造から特定構造を得るための生成過程を説明するための図。

【図18】本実施例における最小候補構造から特定構造を得るための生成過程を説明するための図。

【図19】本実施例における最小候補構造から特定構造を得るための生成過程を説明するための図。

【図20】本実施例における最小候補構造から特定構造を得るための生成過程を説明するための図。

【図21】本実施例における最小候補構造から特定構造を得るための生成過程を説明するための図。

【図22】本実施例における最小候補構造から特定構造を得るための生成過程を説明するための図。

【図23】本実施例における最小候補構造から特定構造を得るための生成過程を説明するための図。

【図24】本実施例における最小候補構造から特定構造を得るための生成過程を説明するための図。

【図25】本実施例における最小候補構造から特定構造を得るための生成過程を説明するための図。

【図26】本実施例における最小候補構造から特定構造を得るための生成過程を説明するための図。

【図27】本実施例における従属子生成子ノードによって有効な編集処理が制限される様子を説明するための図。

【図28】本実施例における従属子生成子ノードによって有効な編集処理が制限される様子を説明するための図。

【図29】本実施例における従属子生成子ノードによって有効な編集処理が制限される様子を説明するための図。

【図30】本実施例における従属子生成子ノードによって有効な編集処理が制限される様子を説明するための図。

【図31】本実施例における従属子生成子ノードによって有効な編集処理が制限される様子を説明するための図。

【図32】本実施例における従属子生成子ノードによって有効な編集処理が制限される様子を説明するための図。

【図33】本実施例における従属子生成子ノードによって有効な編集処理が制限される様子を説明するための図。

【図34】本実施例における従属子生成子ノードによって有効な編集処理が制限される様子を説明するための図。

(13)

特開平5-108631

23

24

図。

【図35】本実施例における従属子生成子ノードによって有効な編集処理が制限される様子を説明するための図。

【図36】従来における特定構造（文書）の編集を説明するための図。

【図37】従来における特定構造（文書）の編集を説明するための図。

【図38】従来における特定構造（文書）の編集を説明するための図。

【図39】従来における特定構造（文書）の編集を説明

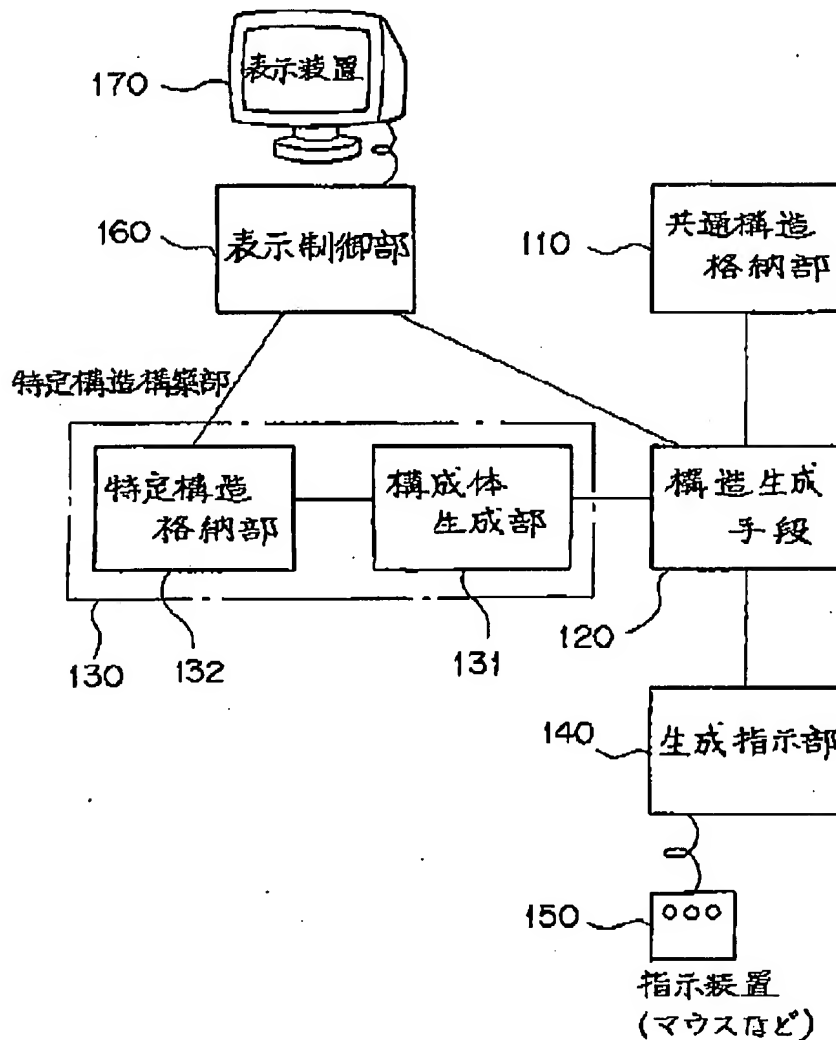
するための図。

【図40】従来における特定構造（文書）の編集を説明するための図。

【符号の説明】

110…共通構造格納部、120…構造生成手段、130…特定構造構築部、131…構成体生成部、132…特定構造格納部、140…生成指示部、150…指示装置、160…表示制御部、170…表示装置、1400、2700…ウィンドウ、1400A、2700A…候補構造表示領域、1400B、2700B…特定構造表示領域。

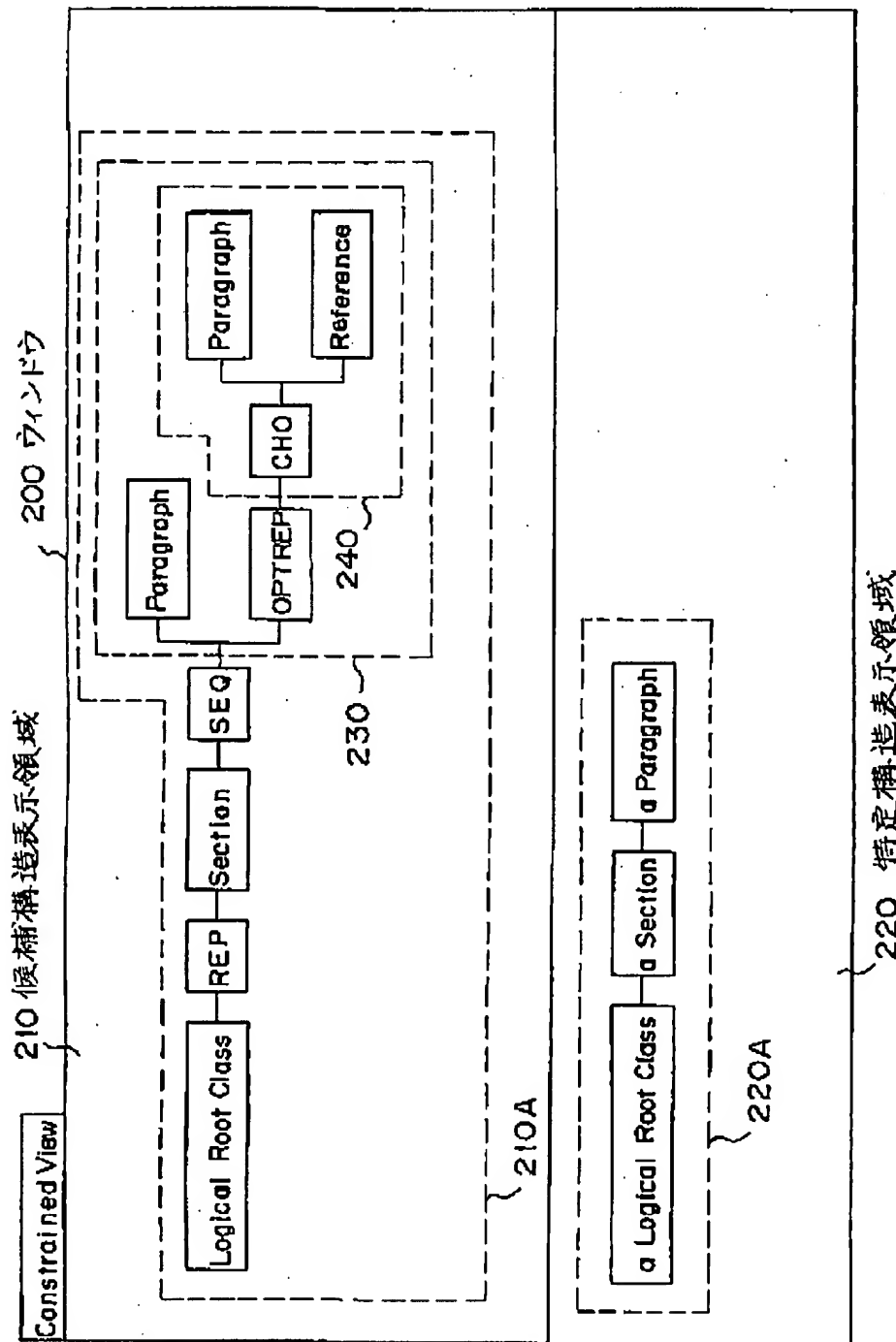
【図1】



(14)

特開平5-108631

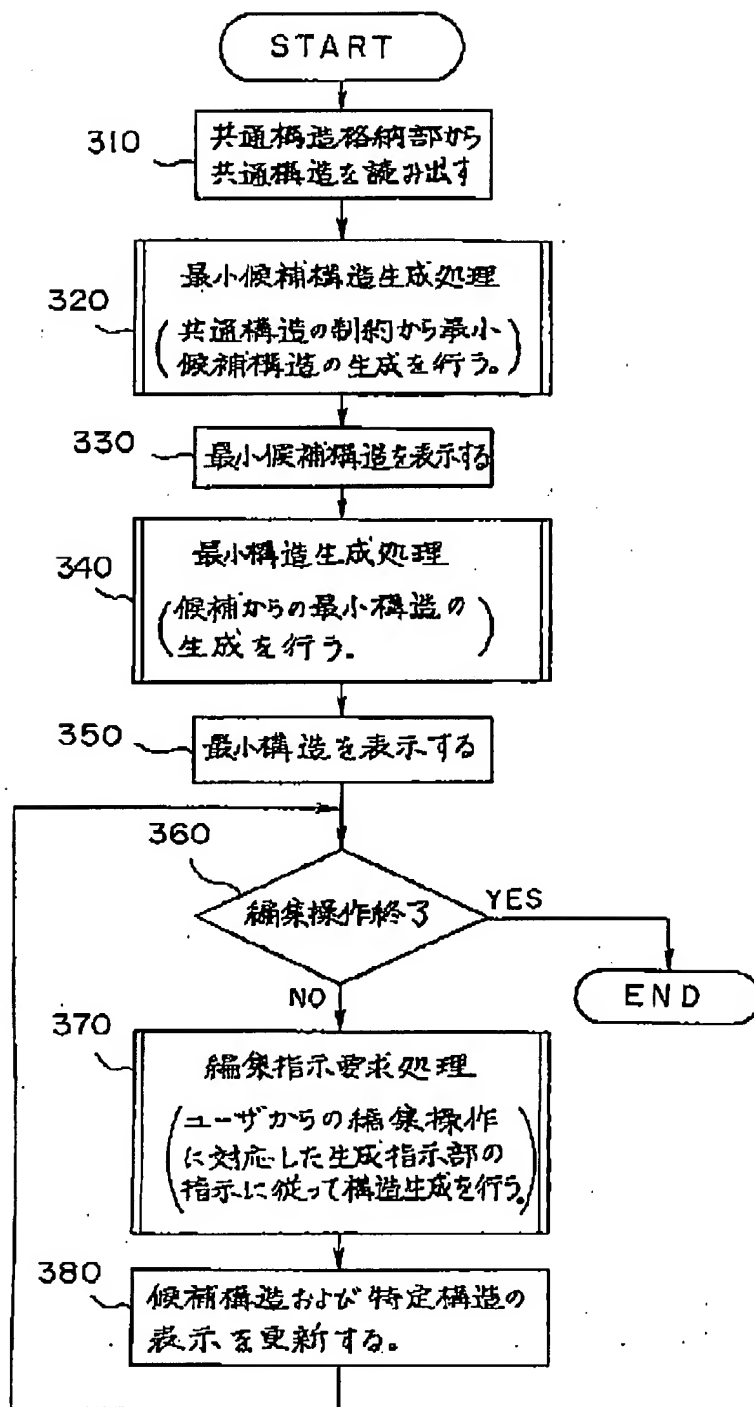
【図2】



(15)

特開平5-108631

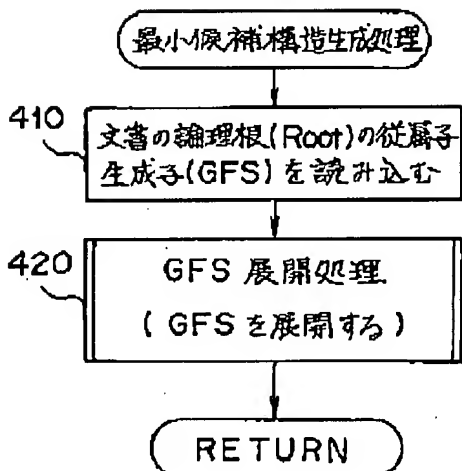
【図3】



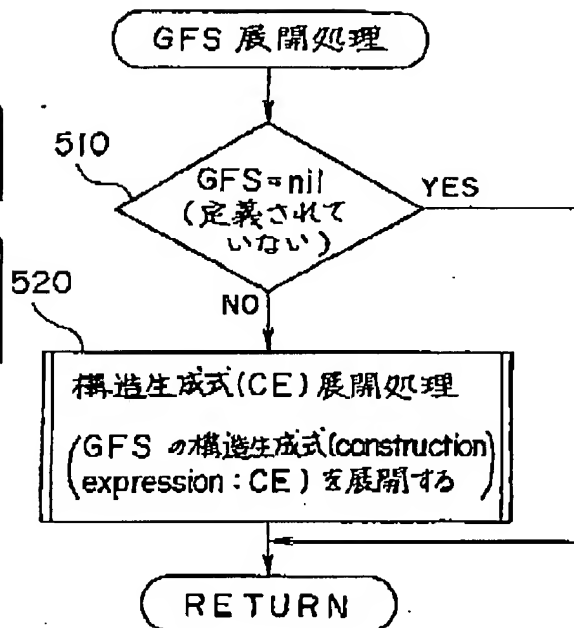
(16)

特開平5-108631

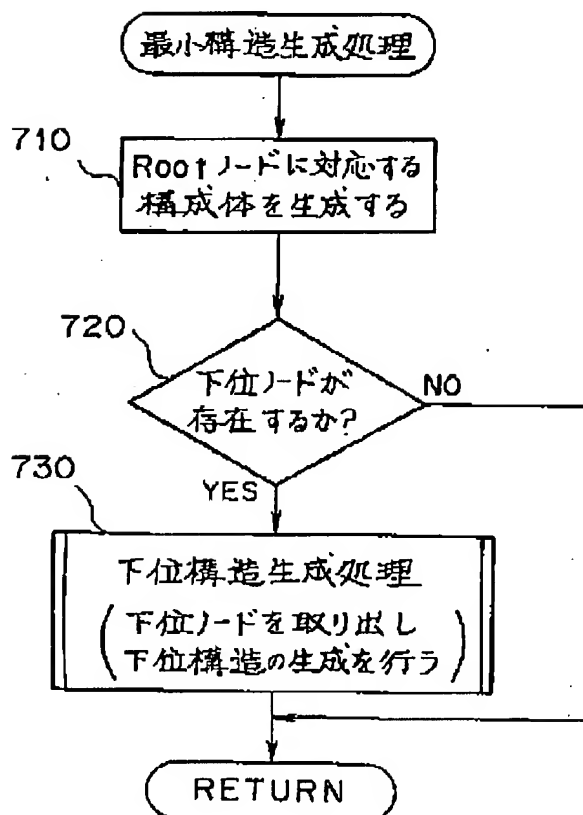
【図4】



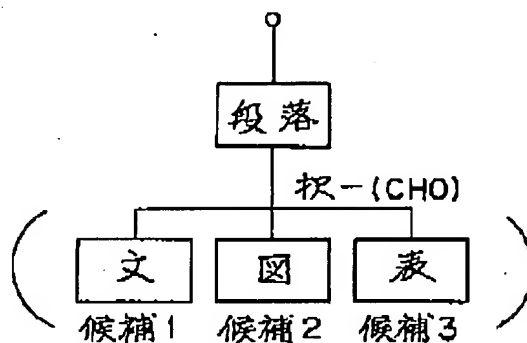
【図5】



【図7】



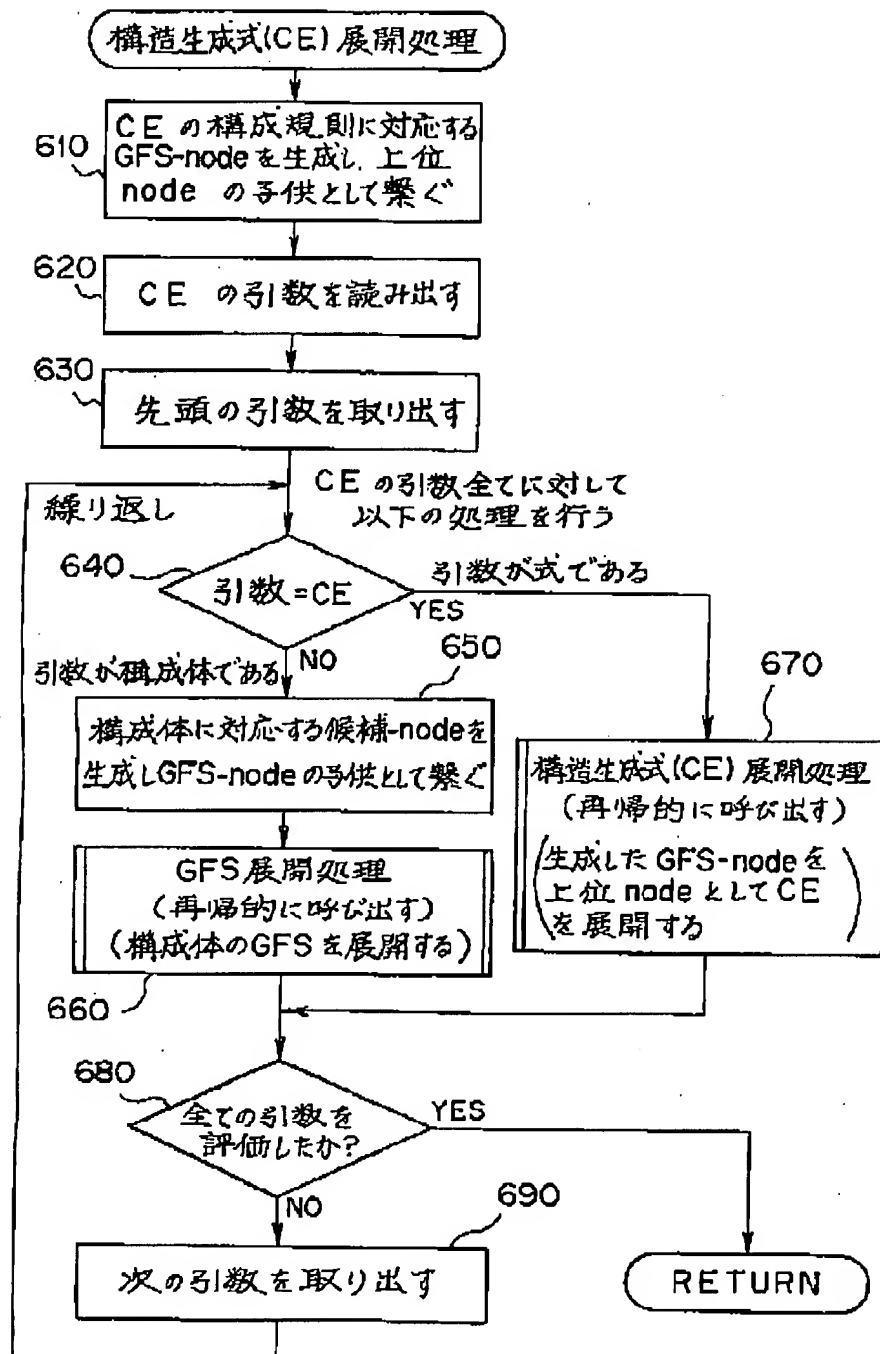
【図39】



(17)

特開平5-108631

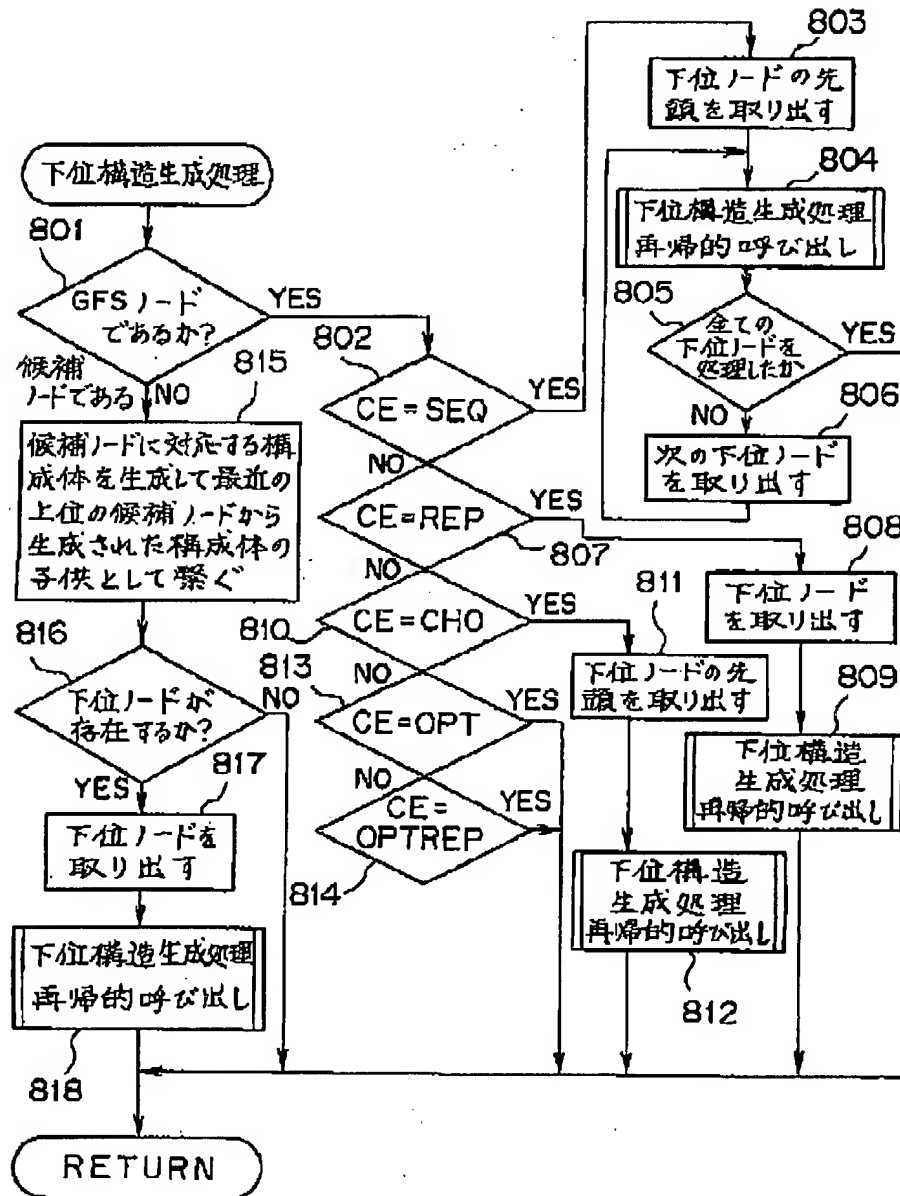
【図6】



(18)

特開平5-108631

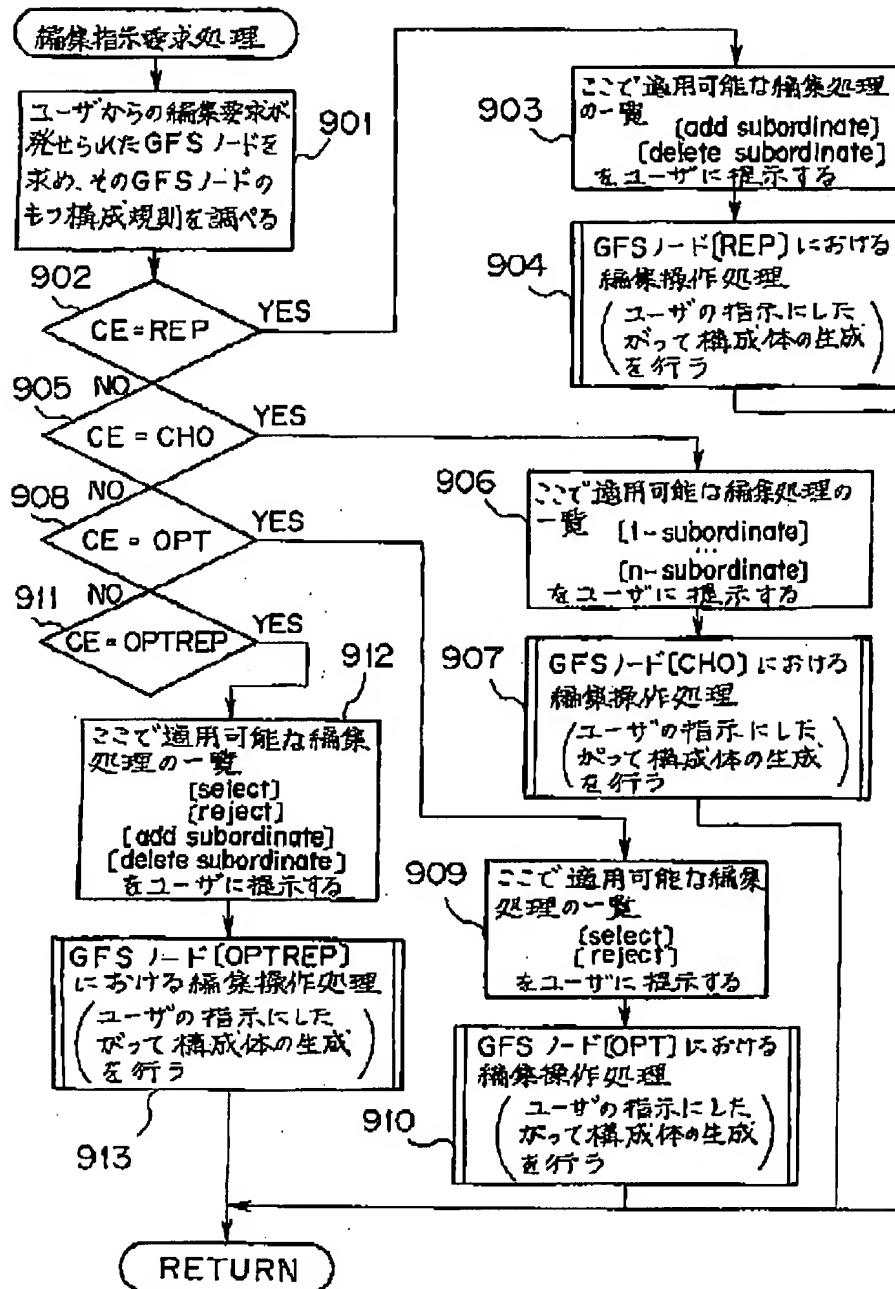
【図8】



(19)

特開平5-108631

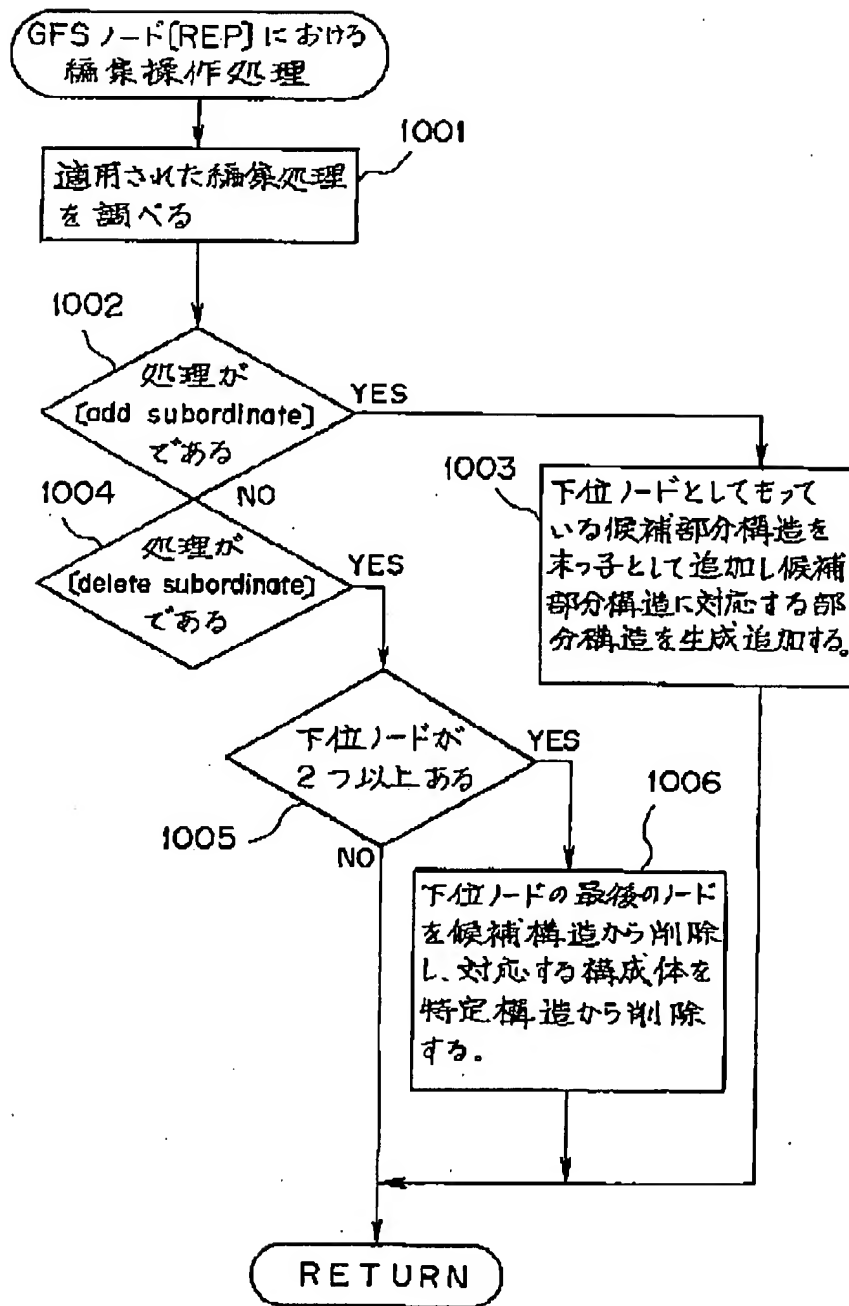
【図9】



(20)

特開平5-108631

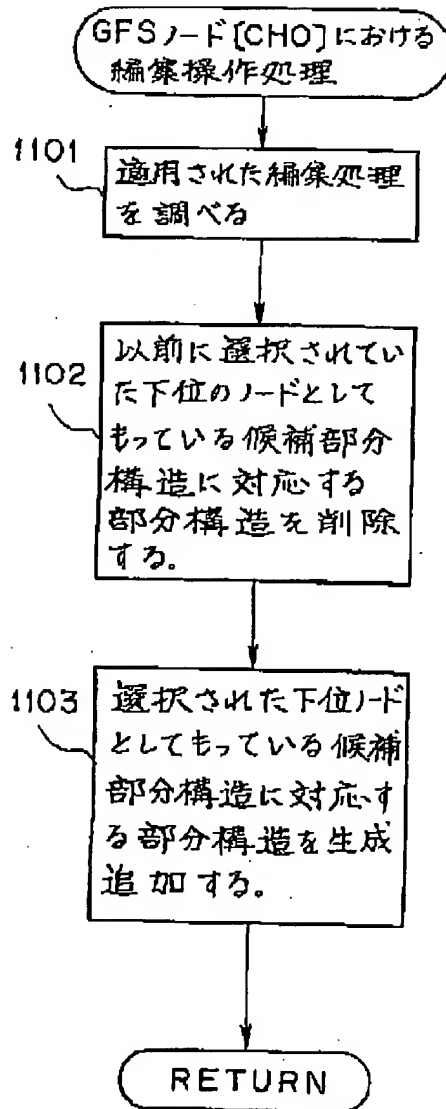
【図10】



(21)

特開平5-108631

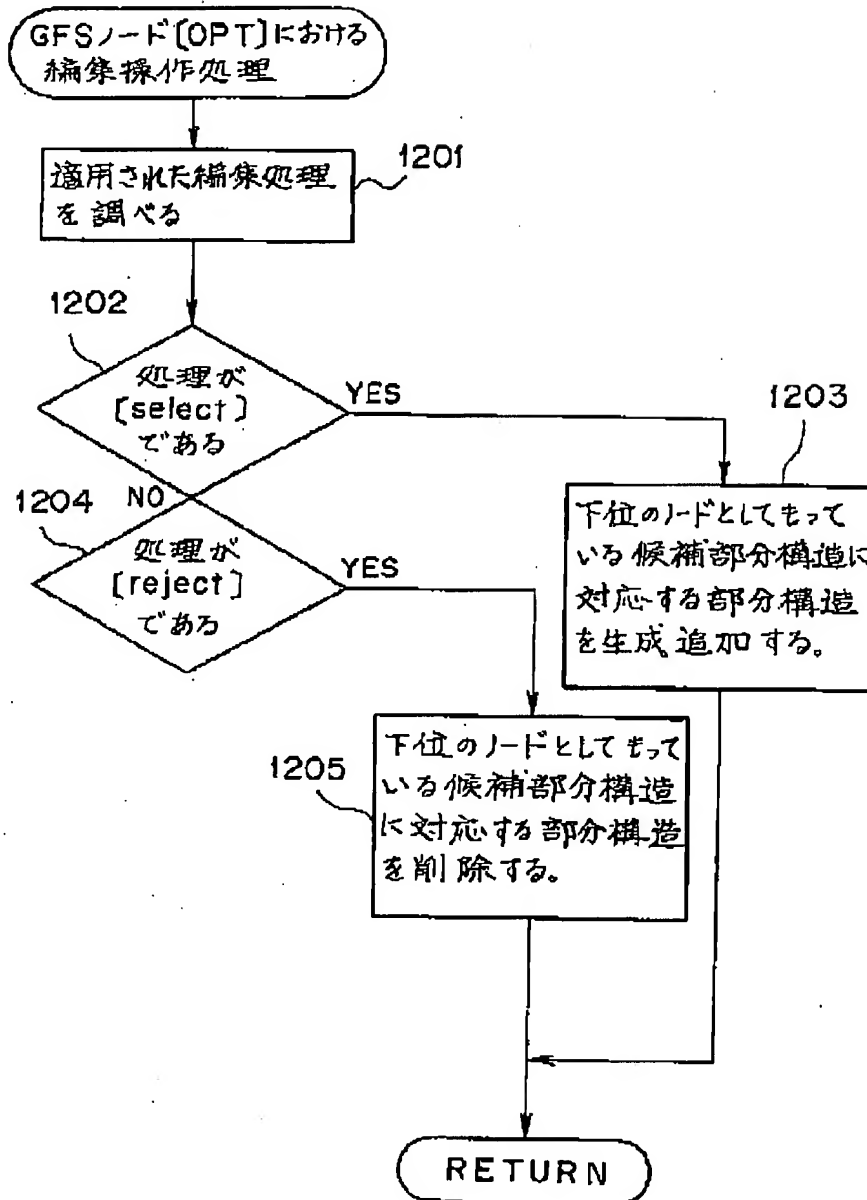
【図11】



(22)

特開平5-108631

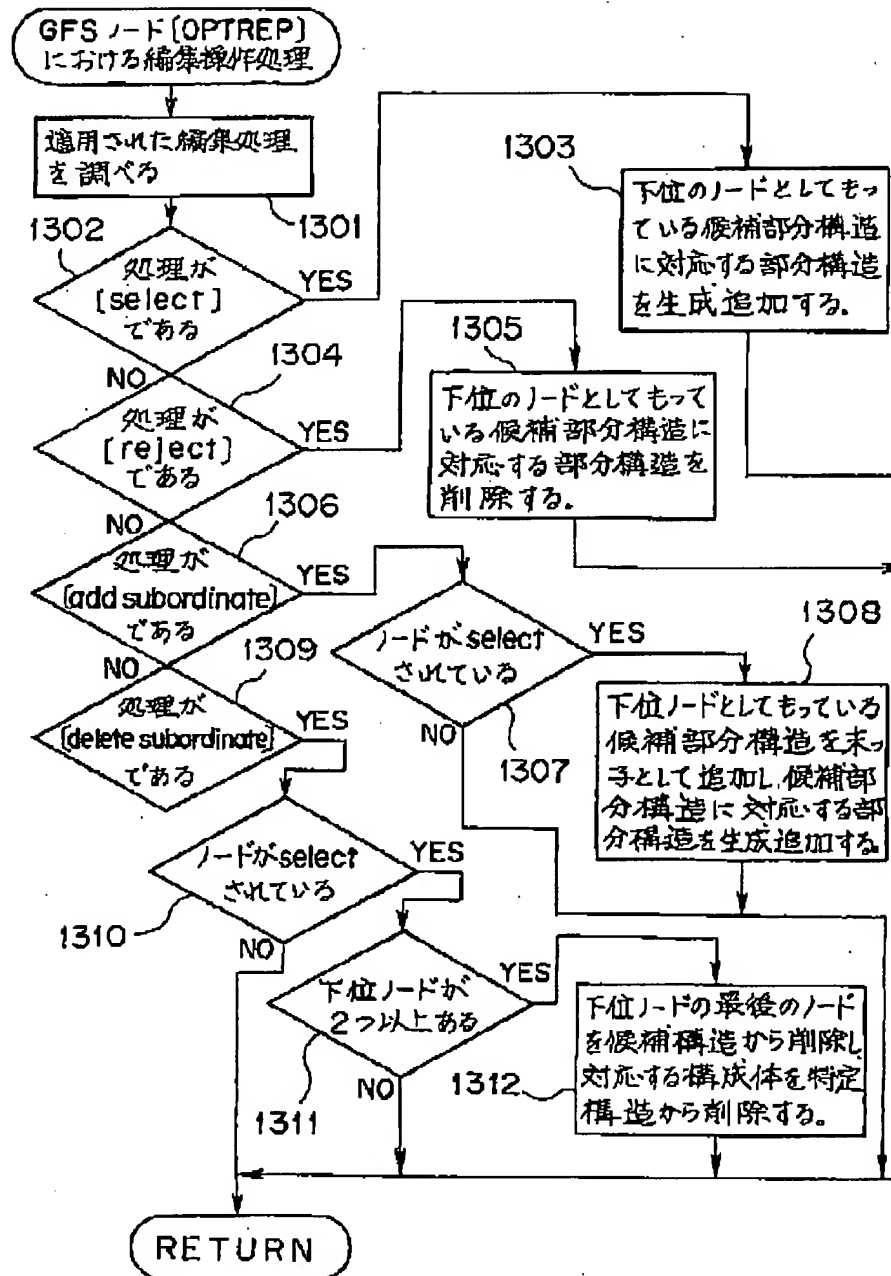
【図12】



(23)

特開平5-108631

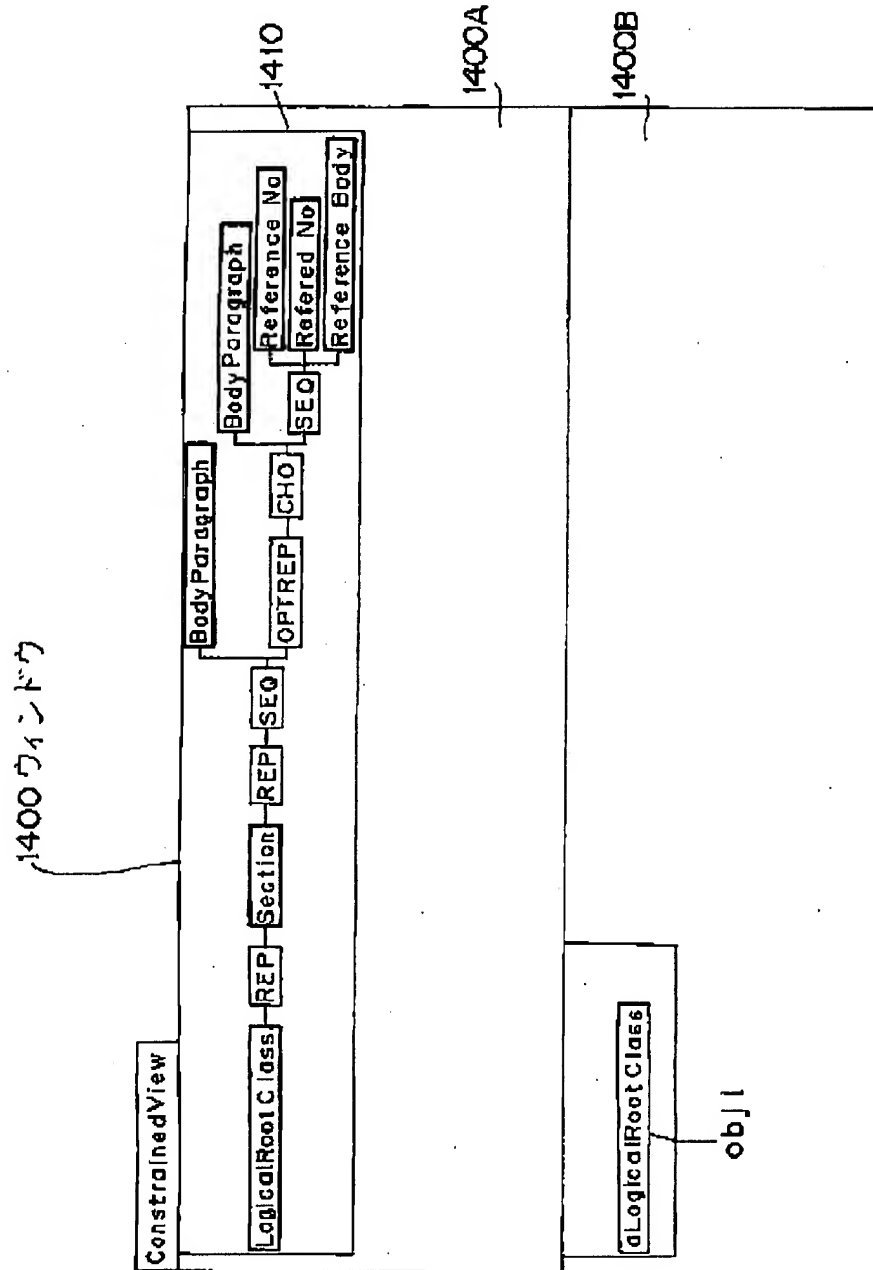
【図13】



(24)

特開平5-108631

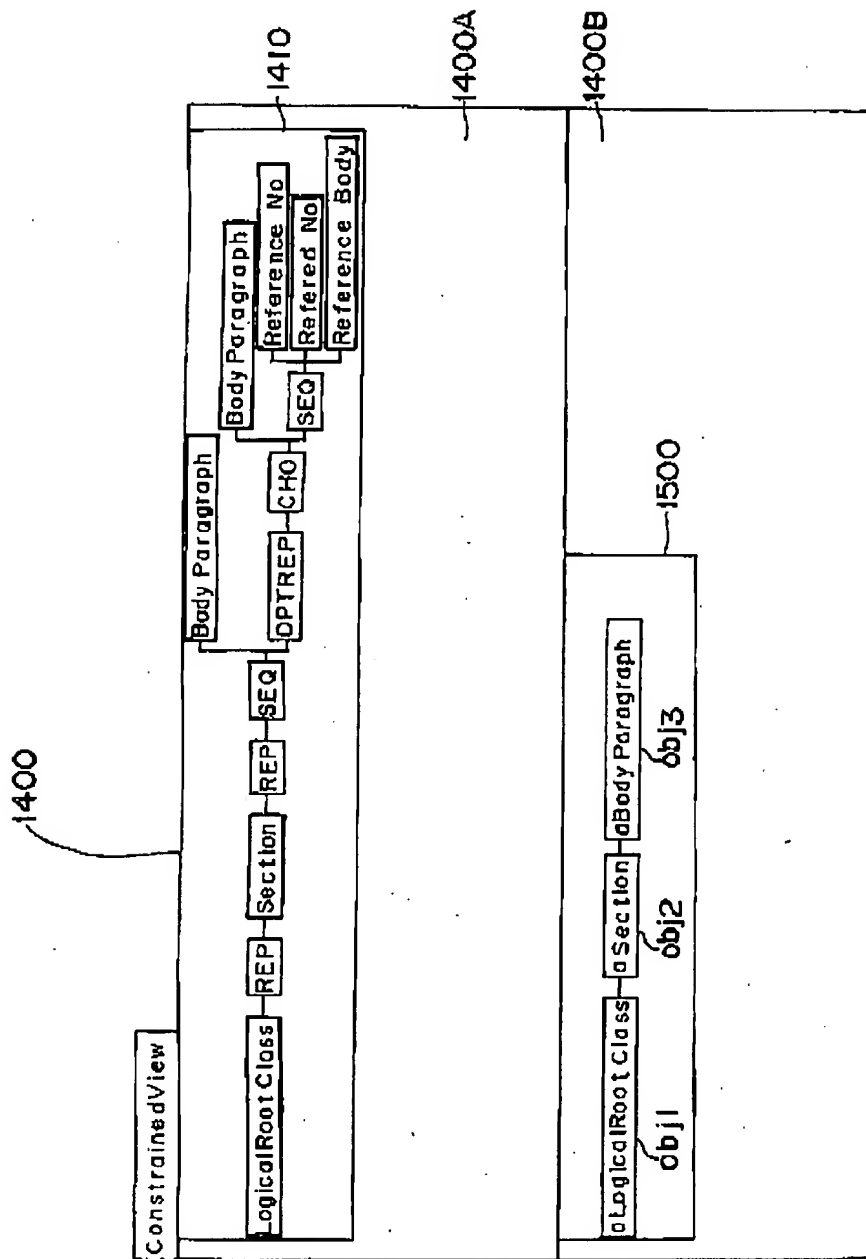
【図14】



(25)

特開平5-108631

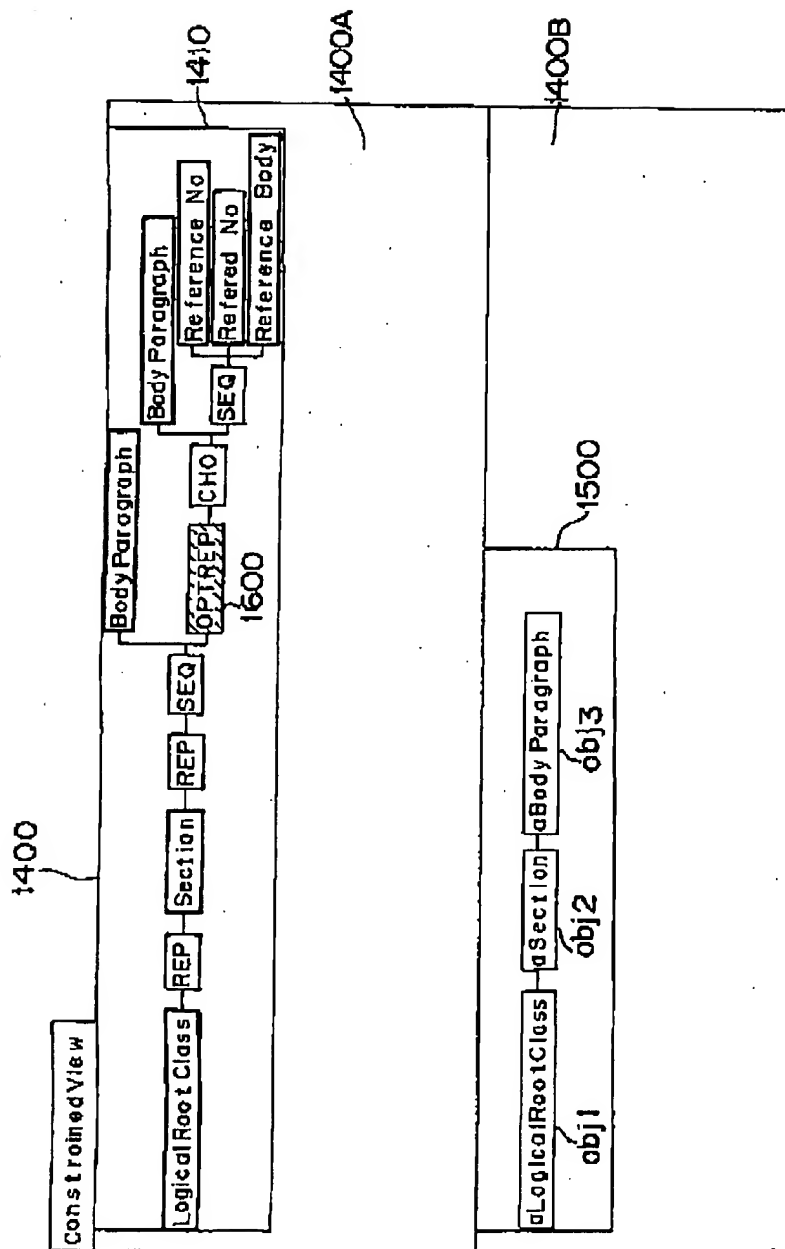
【図15】



(26)

特開平5-108631

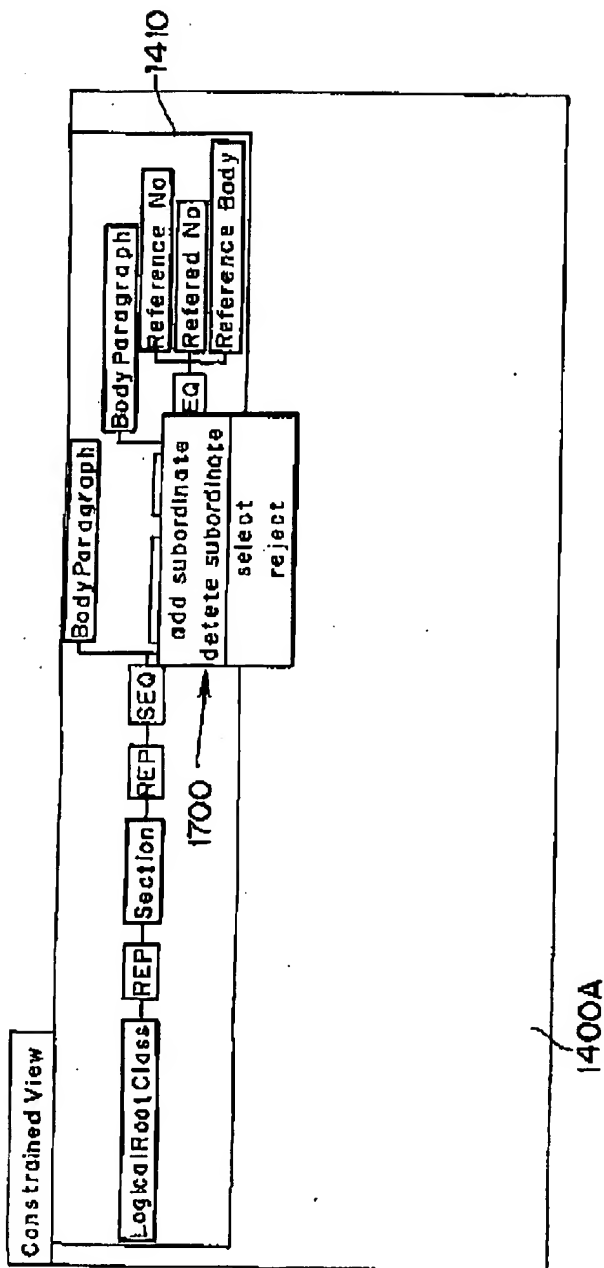
【图 16】



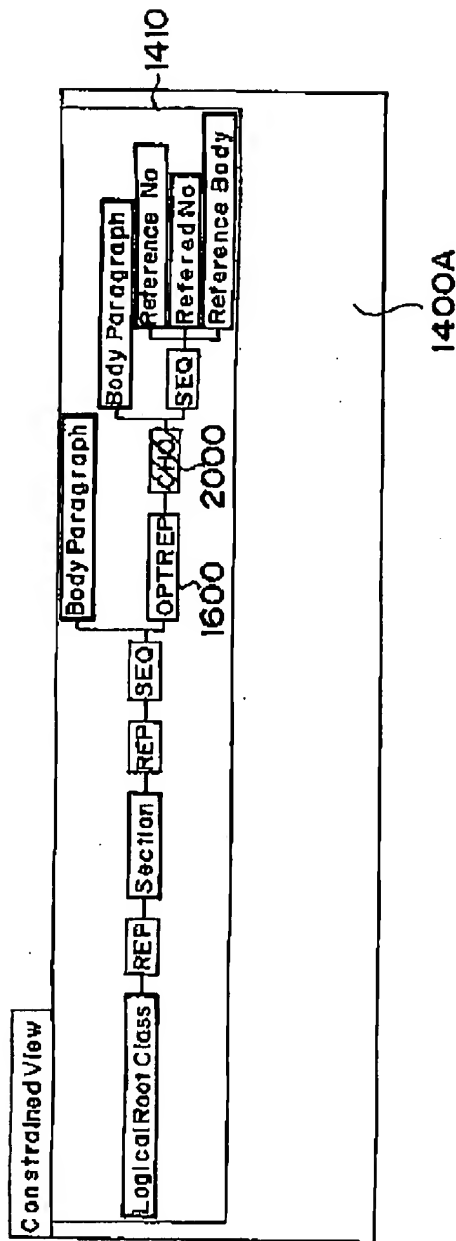
(27)

特開平5-108631

【図17】



【図20】

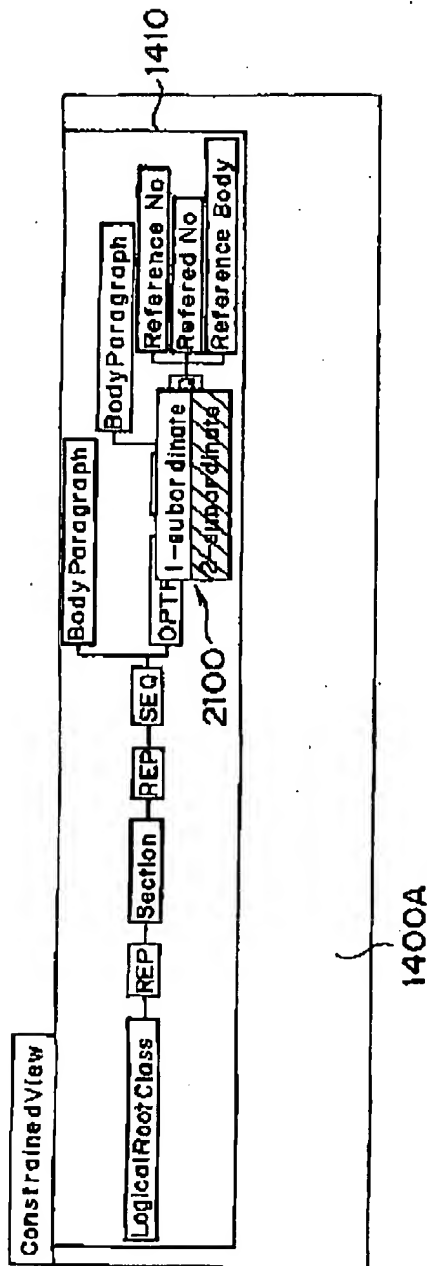
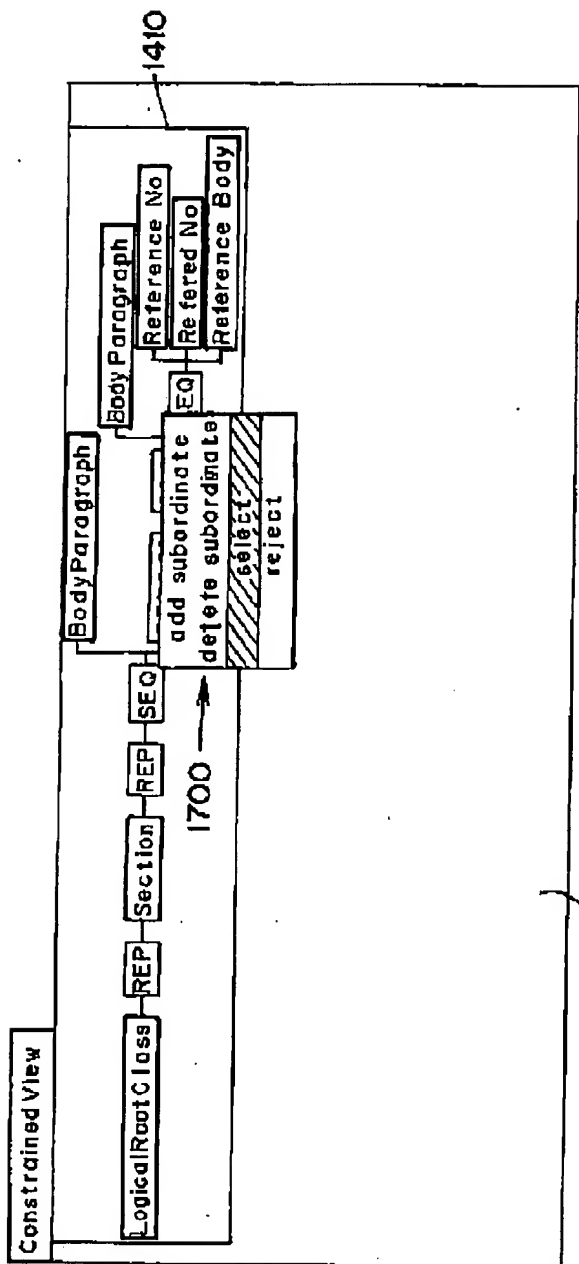


(28)

特開平5-108631

【図18】

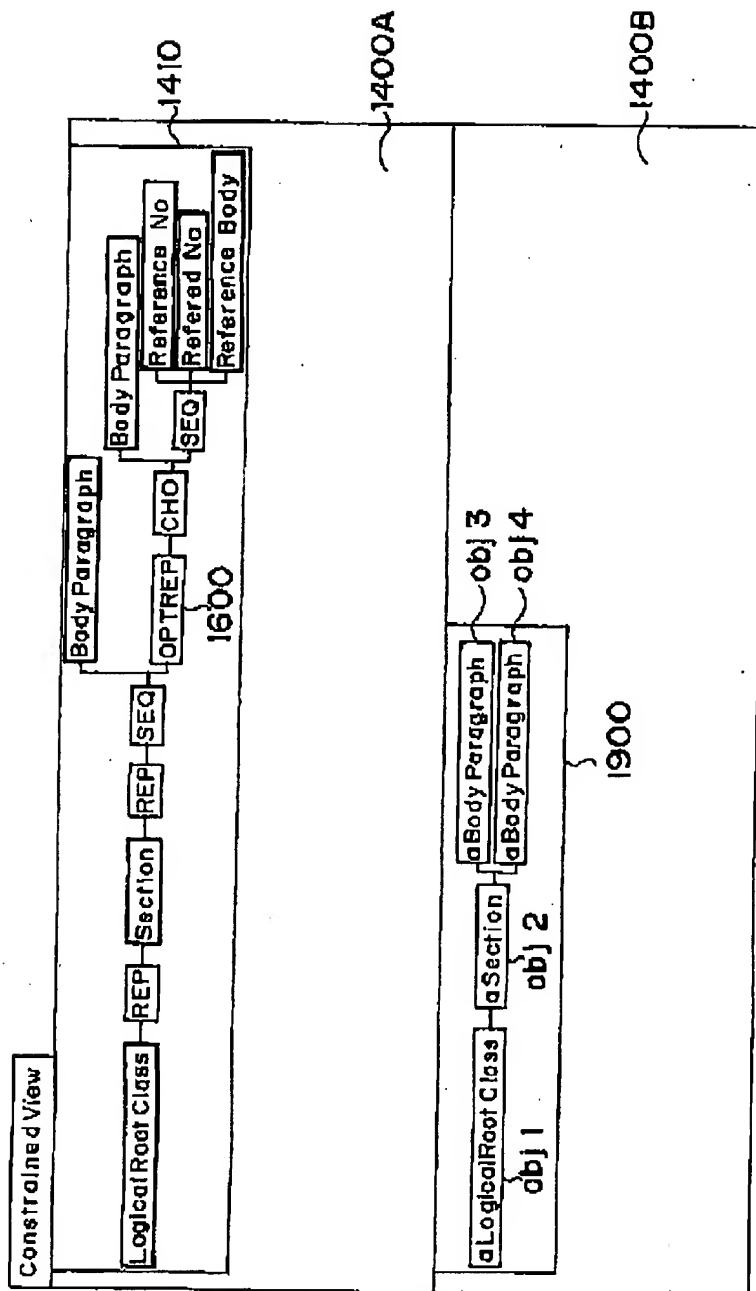
【図21】



(29)

特開平5-108631

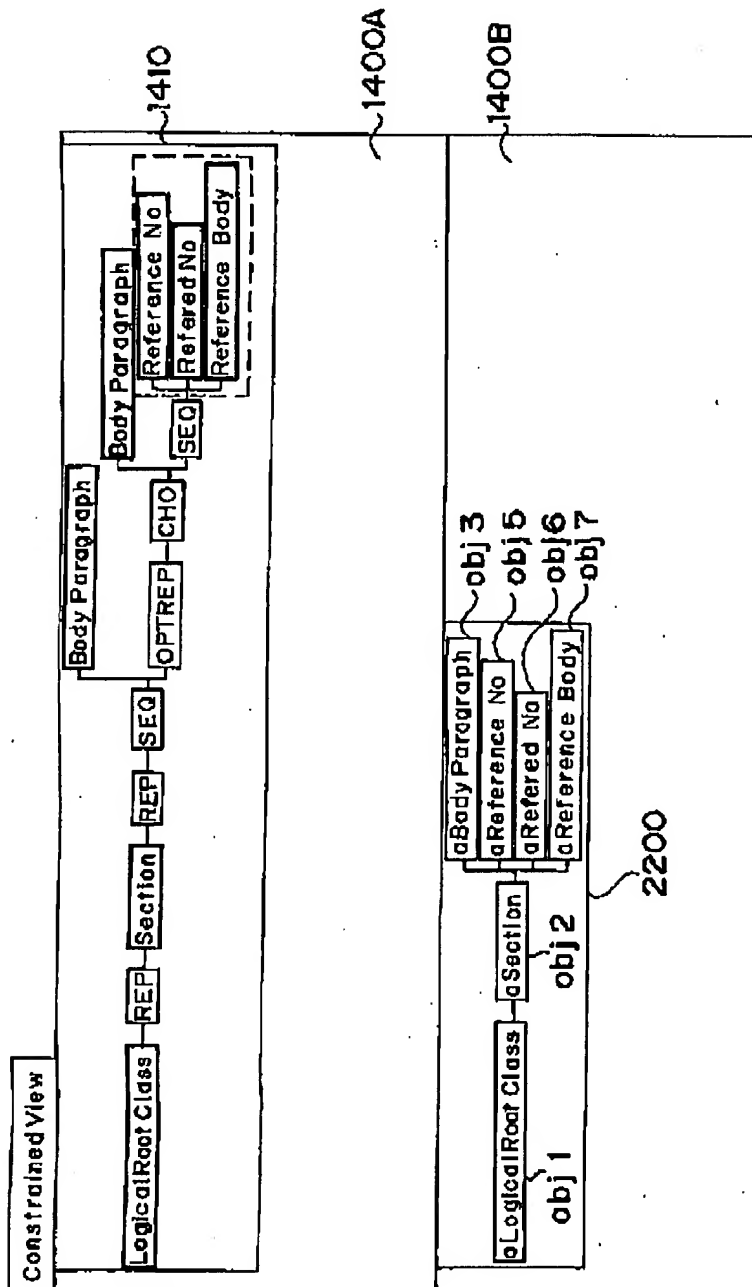
【図19】



(30)

特開平5-108631

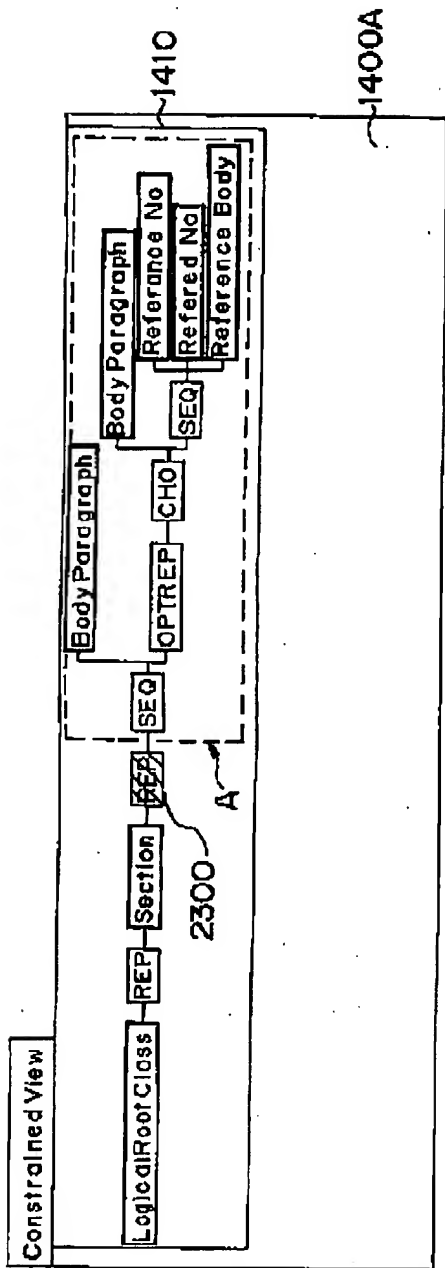
【図22】



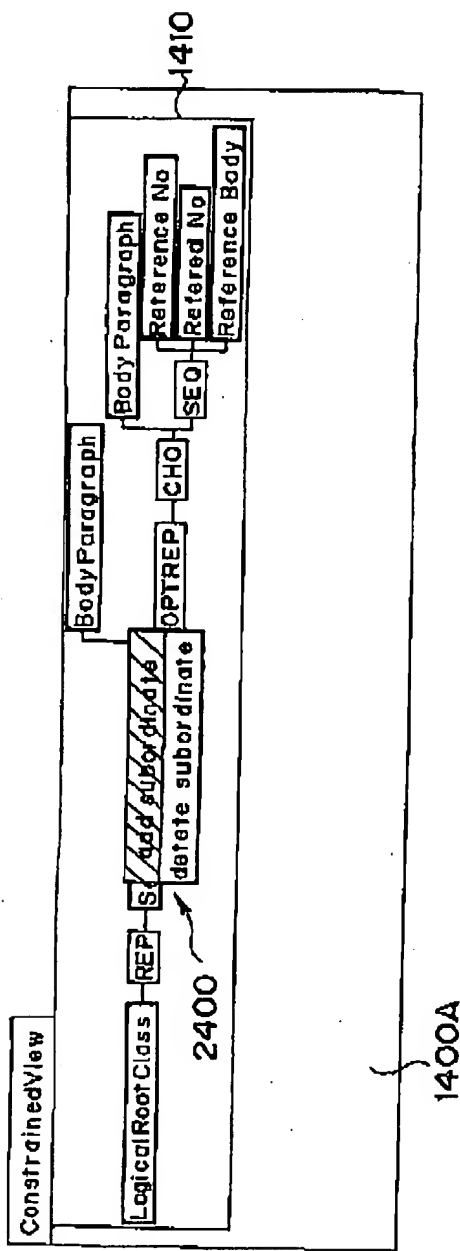
(31)

特開平5-108631

【図23】



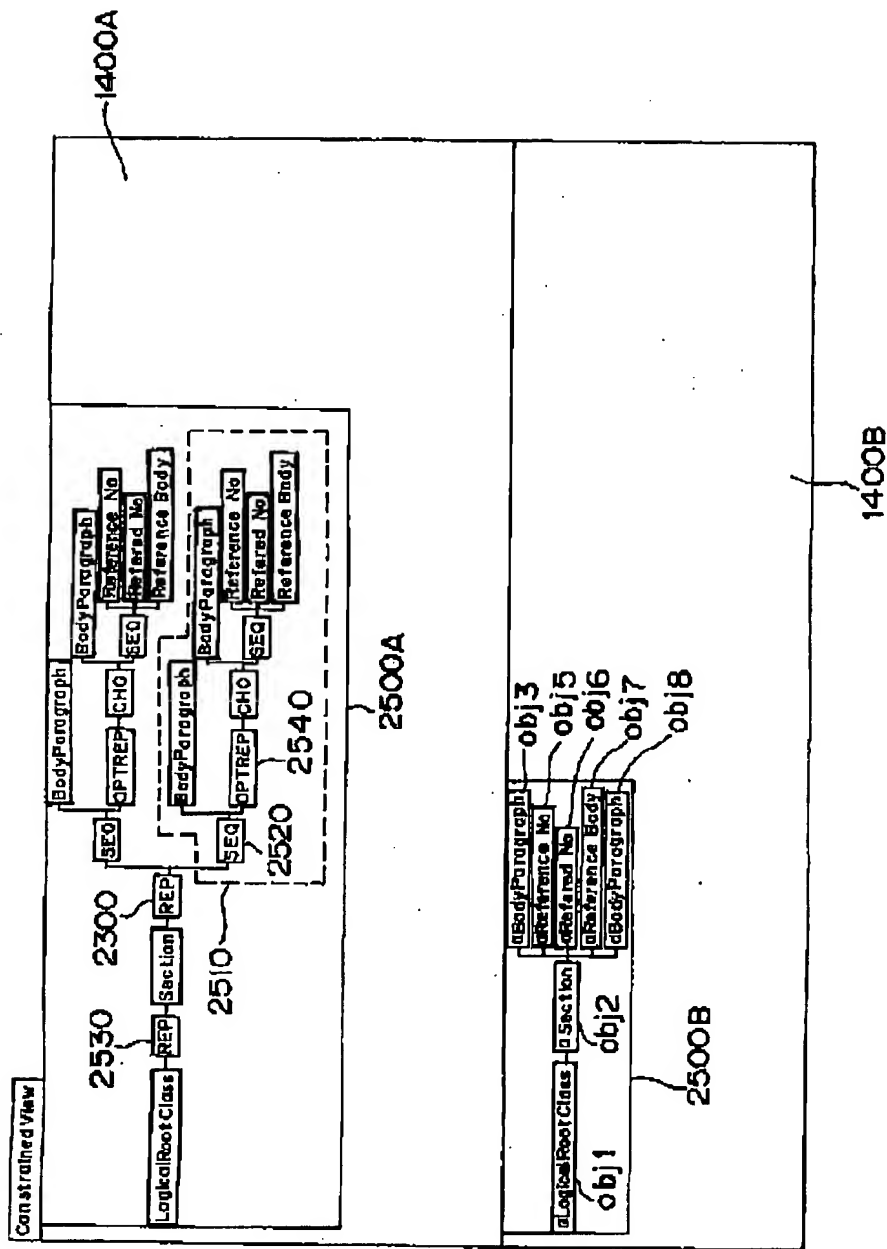
【図24】



(32)

特開平5-108631

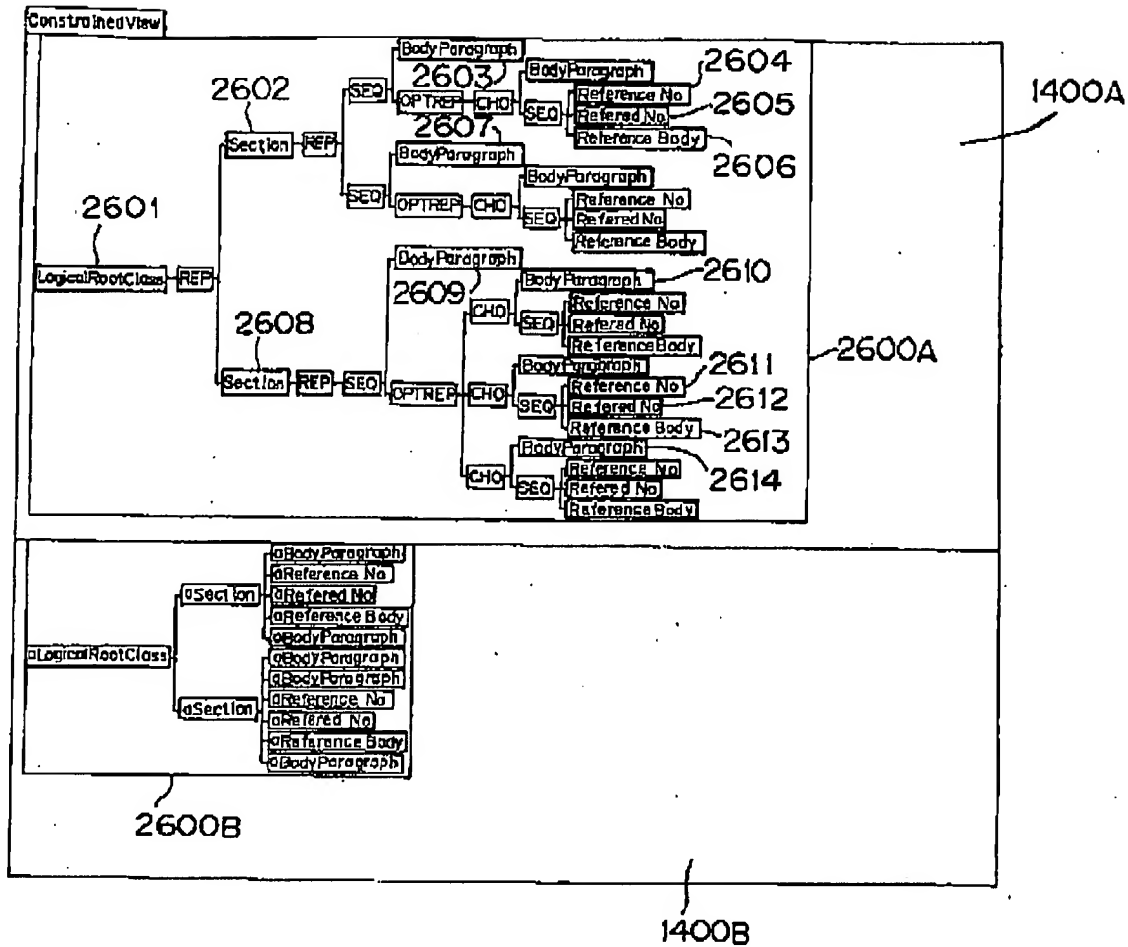
【図25】



(33)

特開平5-108631

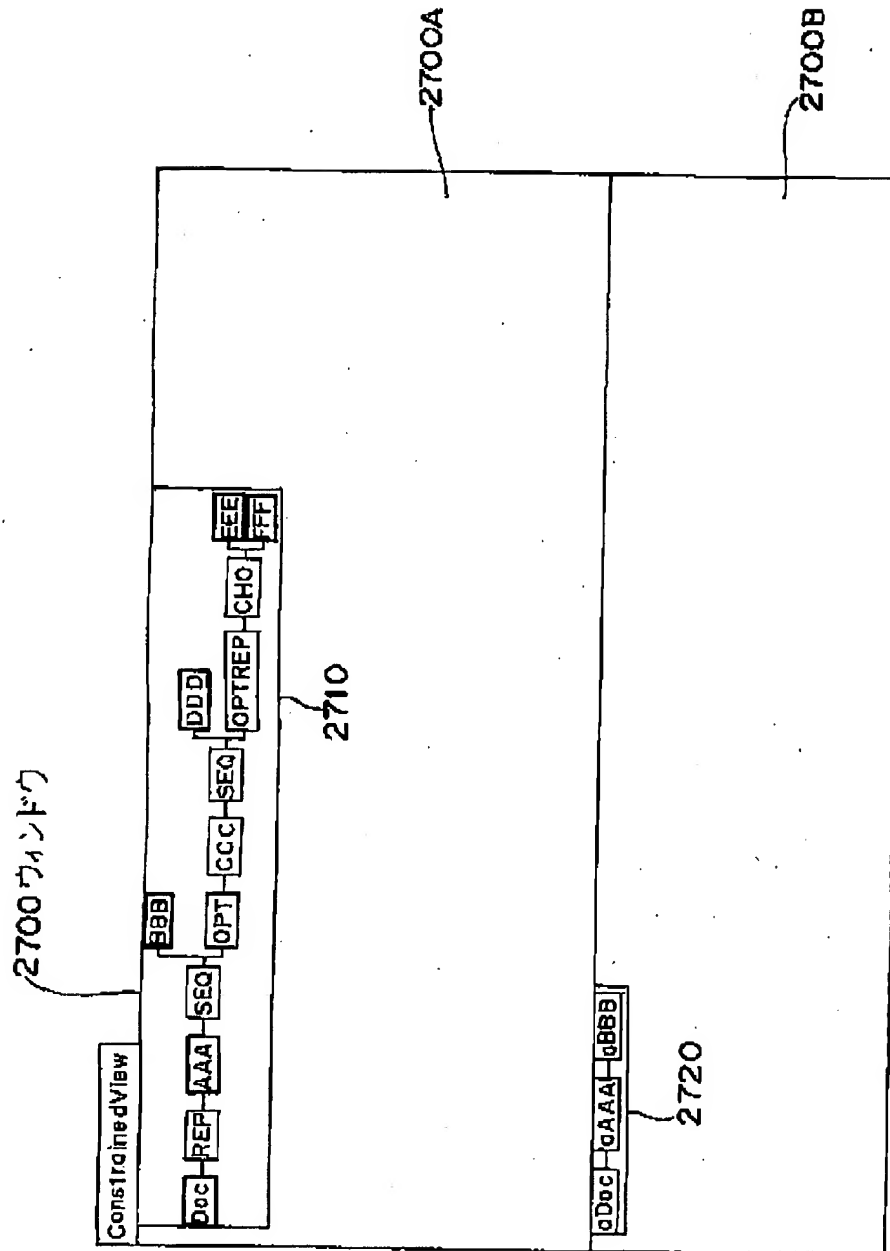
【図26】



(34)

特開平5-108631

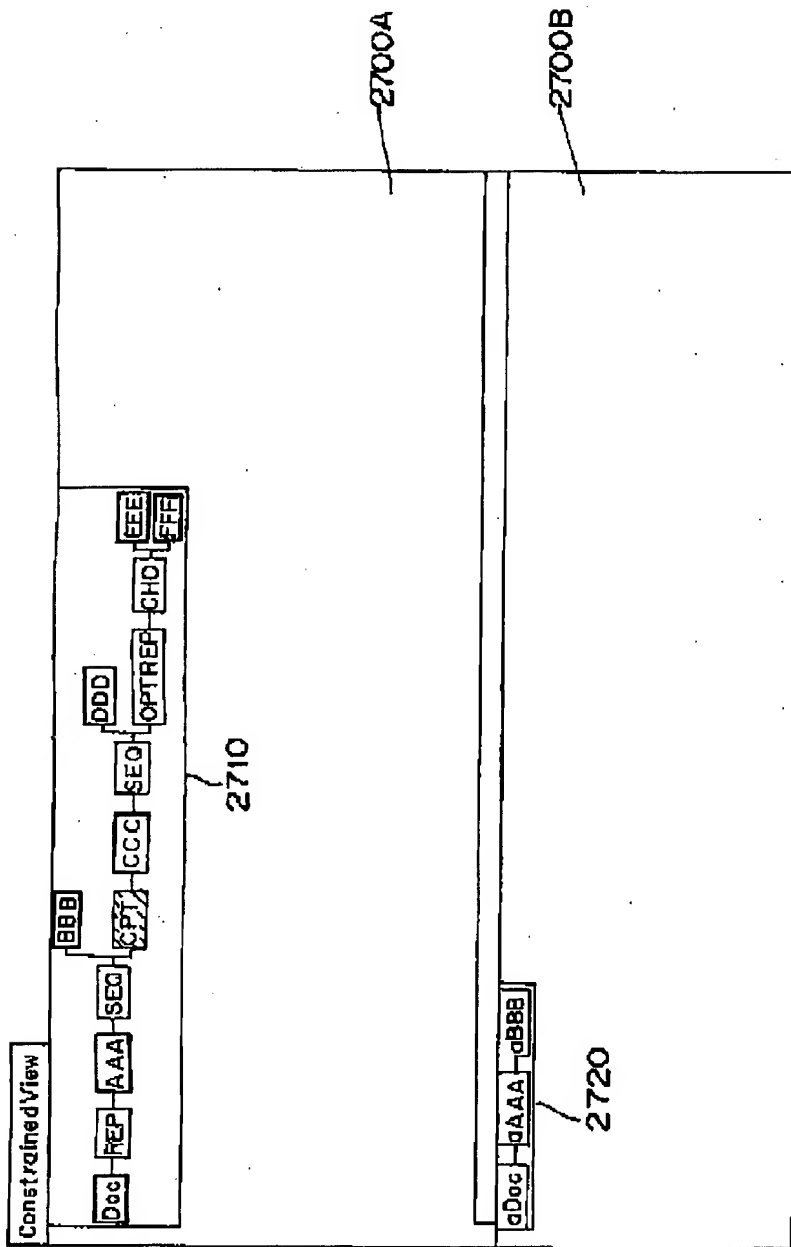
【図27】



(35)

特開平5-108631

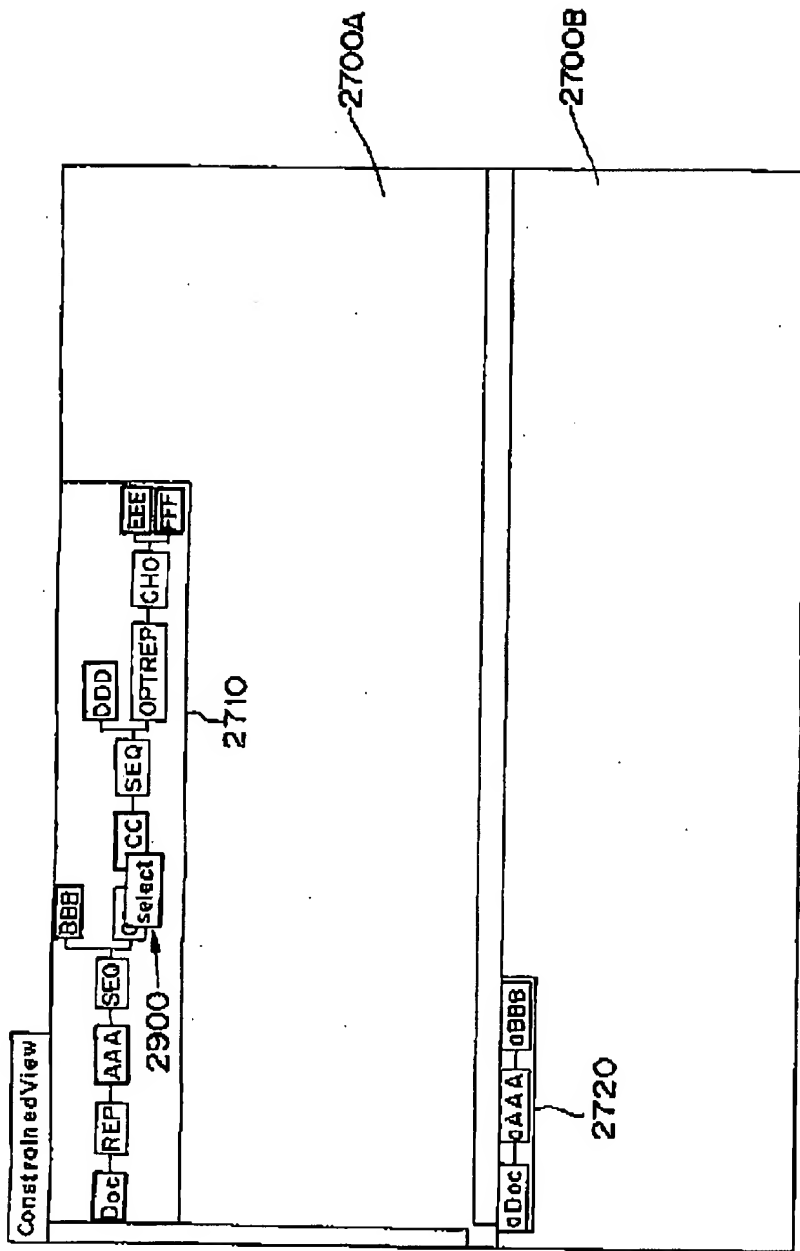
[図28]



(36)

特開平5-108631

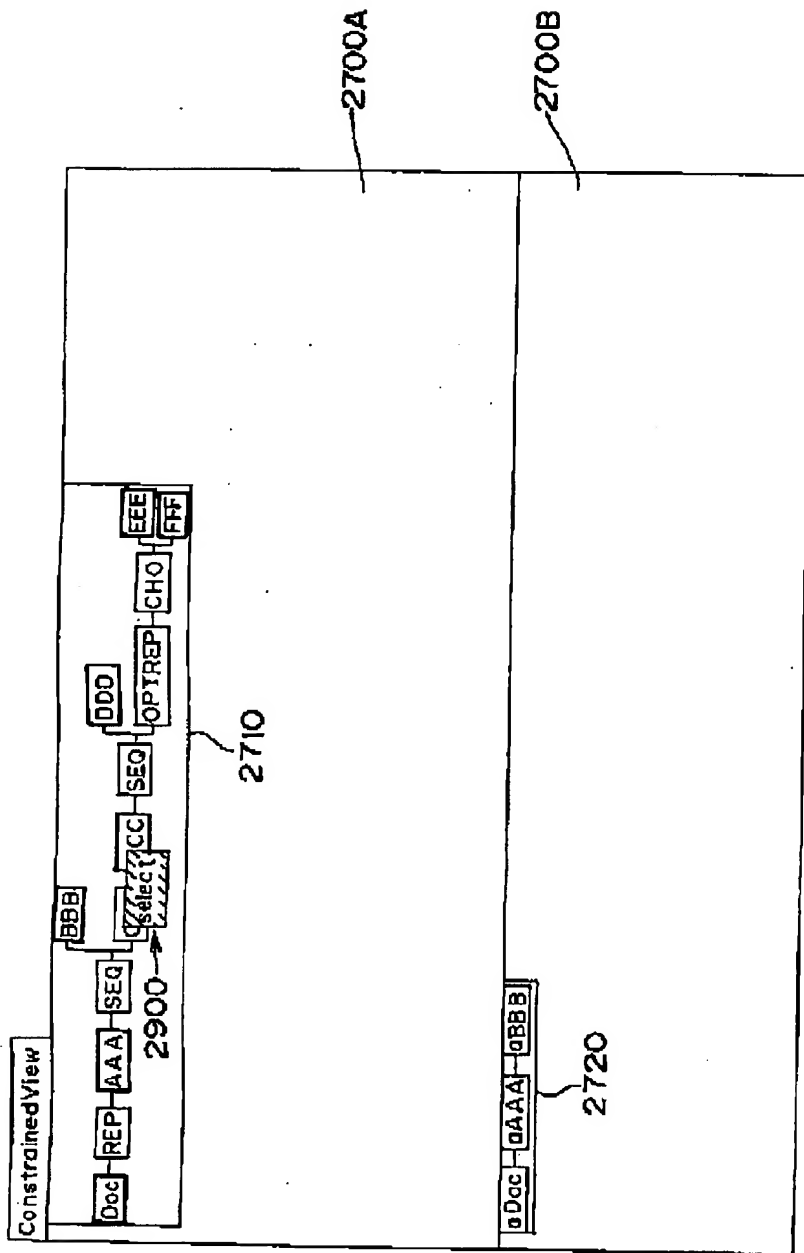
【図29】



(37)

特開平5-108631

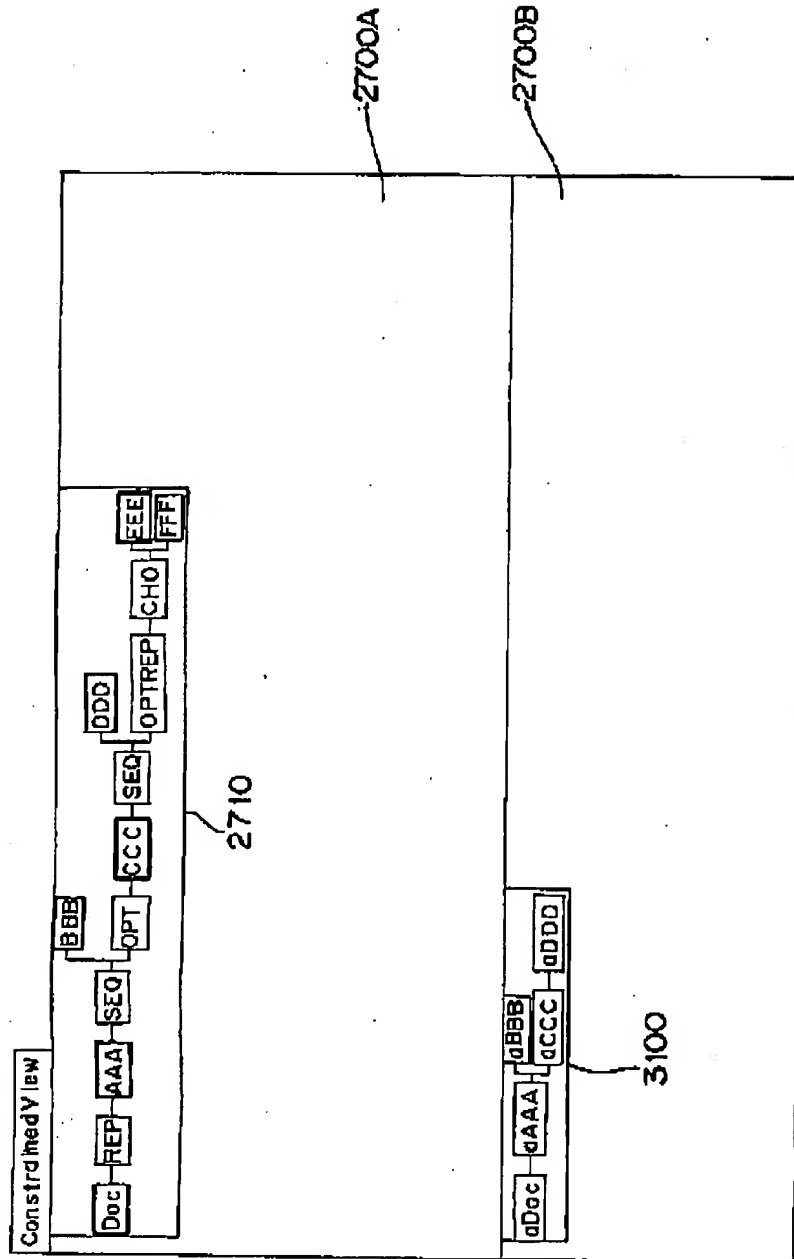
【図30】



(38)

特開平5-108631

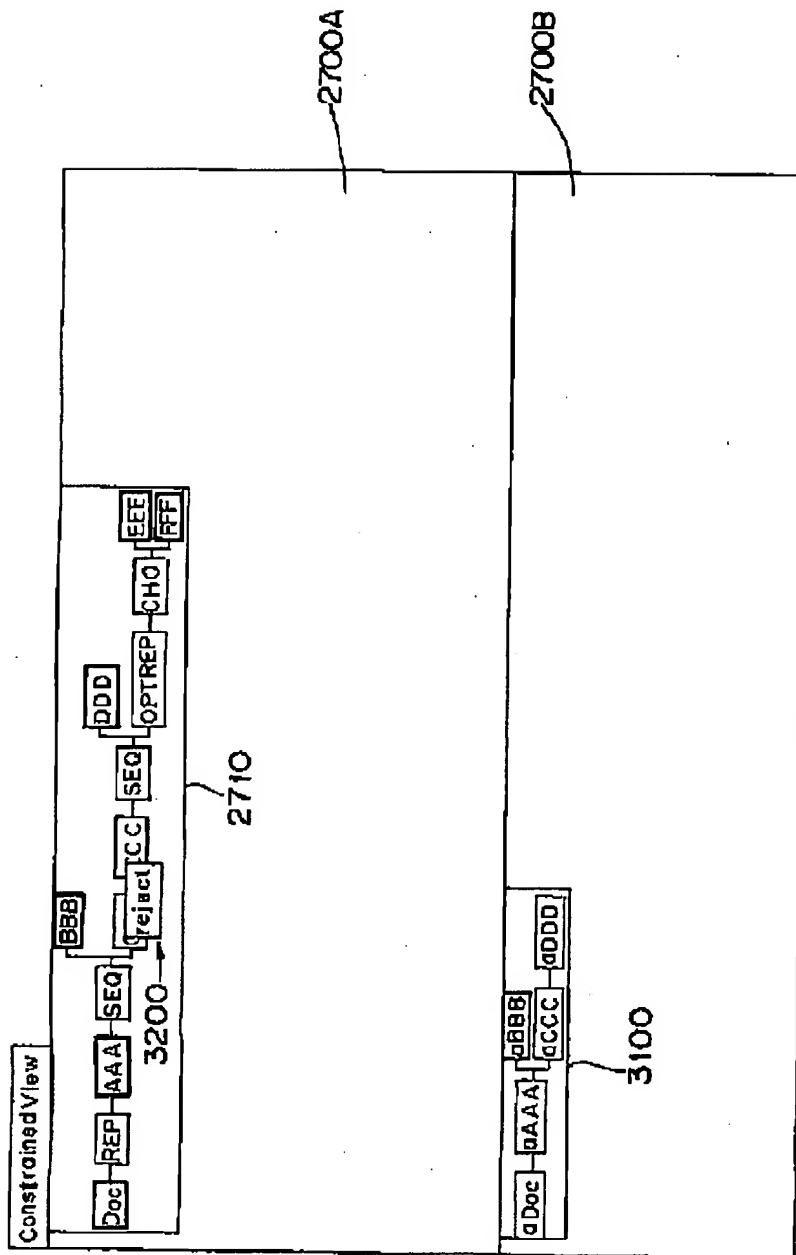
【図31】



(39)

特開平5-108631

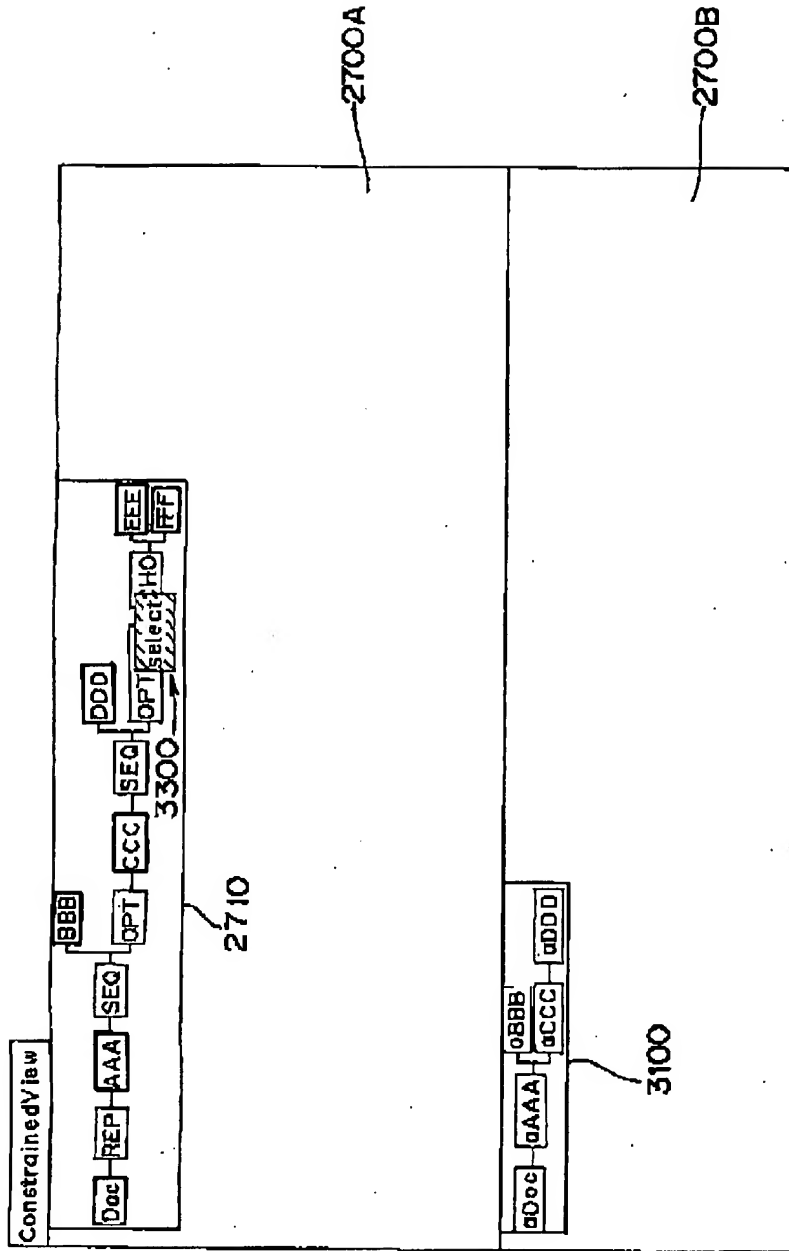
[図32]



(40)

特開平5-108631

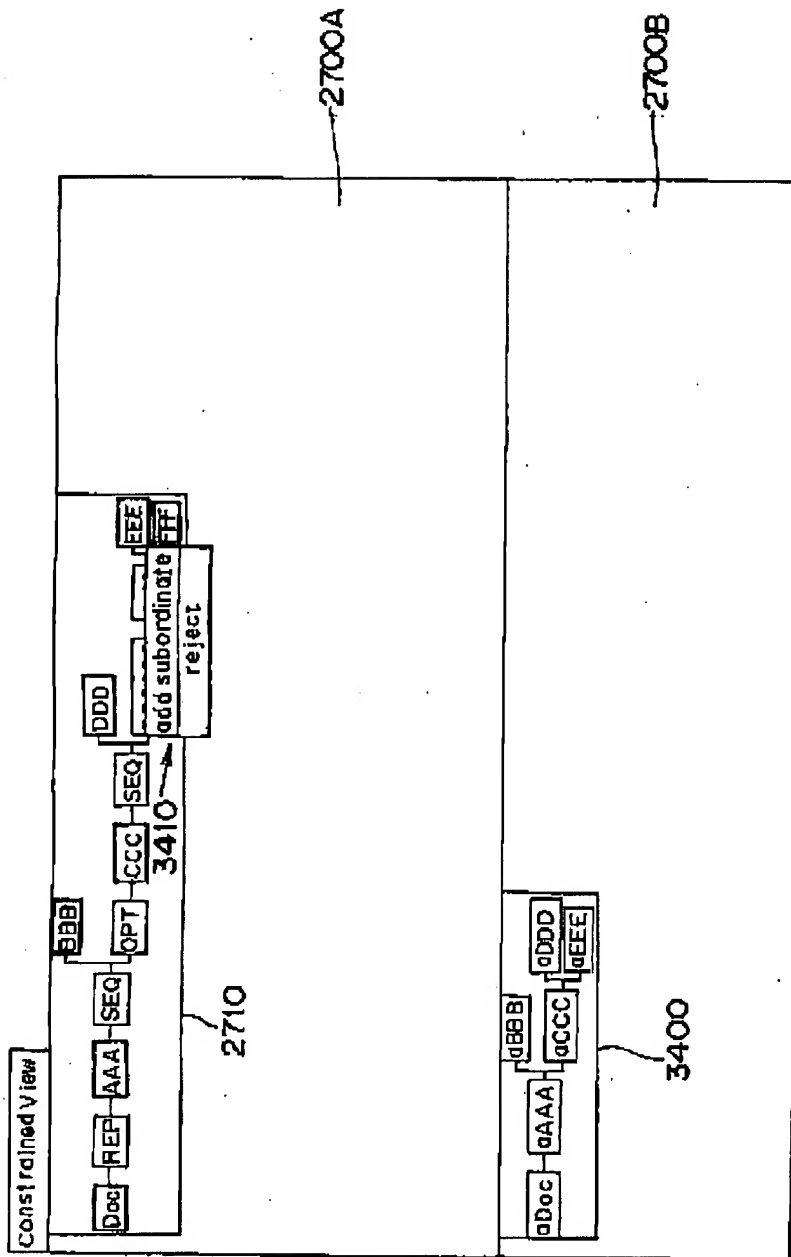
【図33】



(41)

特開平5-108631

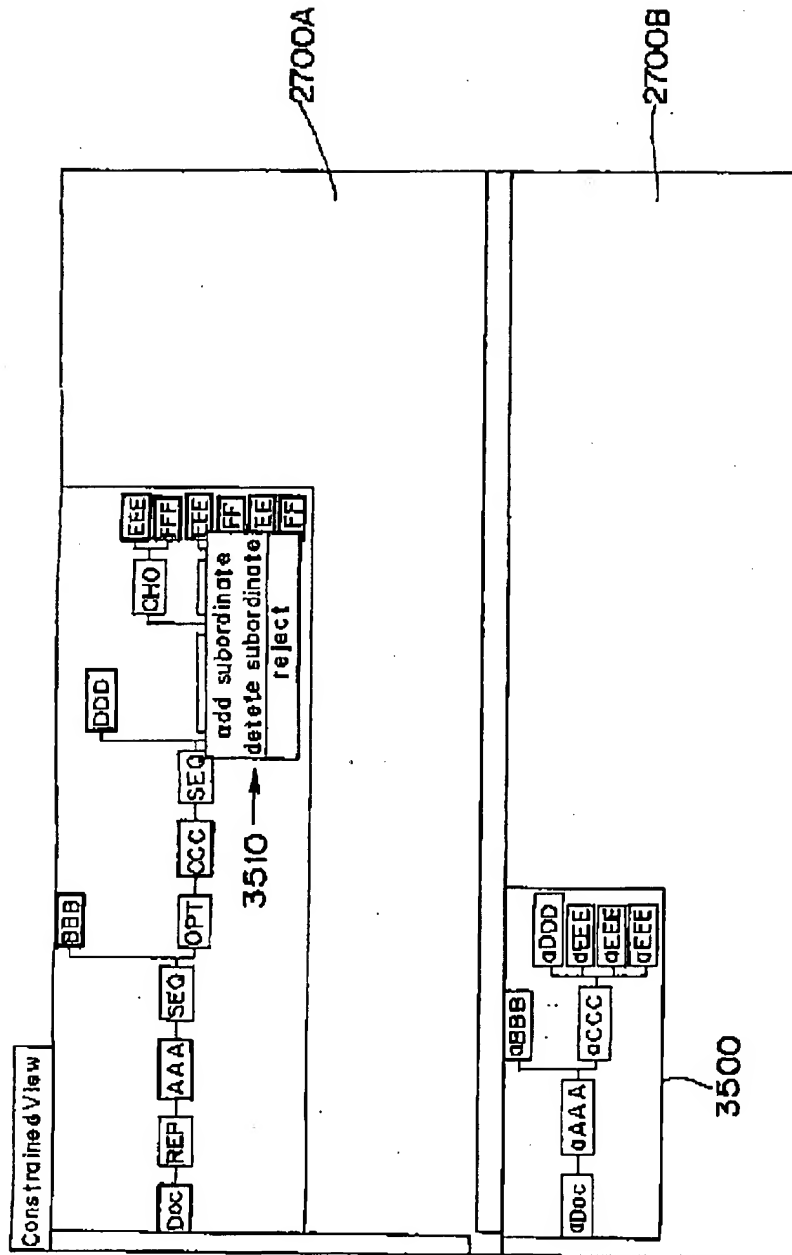
【図34】



(42)

特開平5-108631

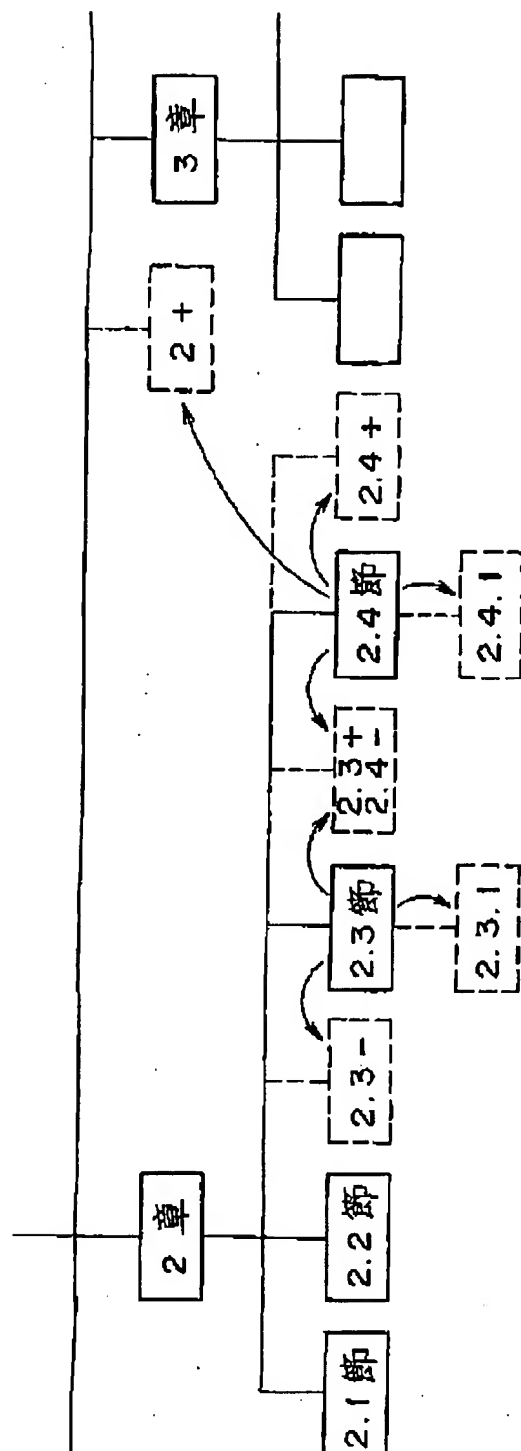
【図35】



(43)

特開平5-108631

【图 3-6】



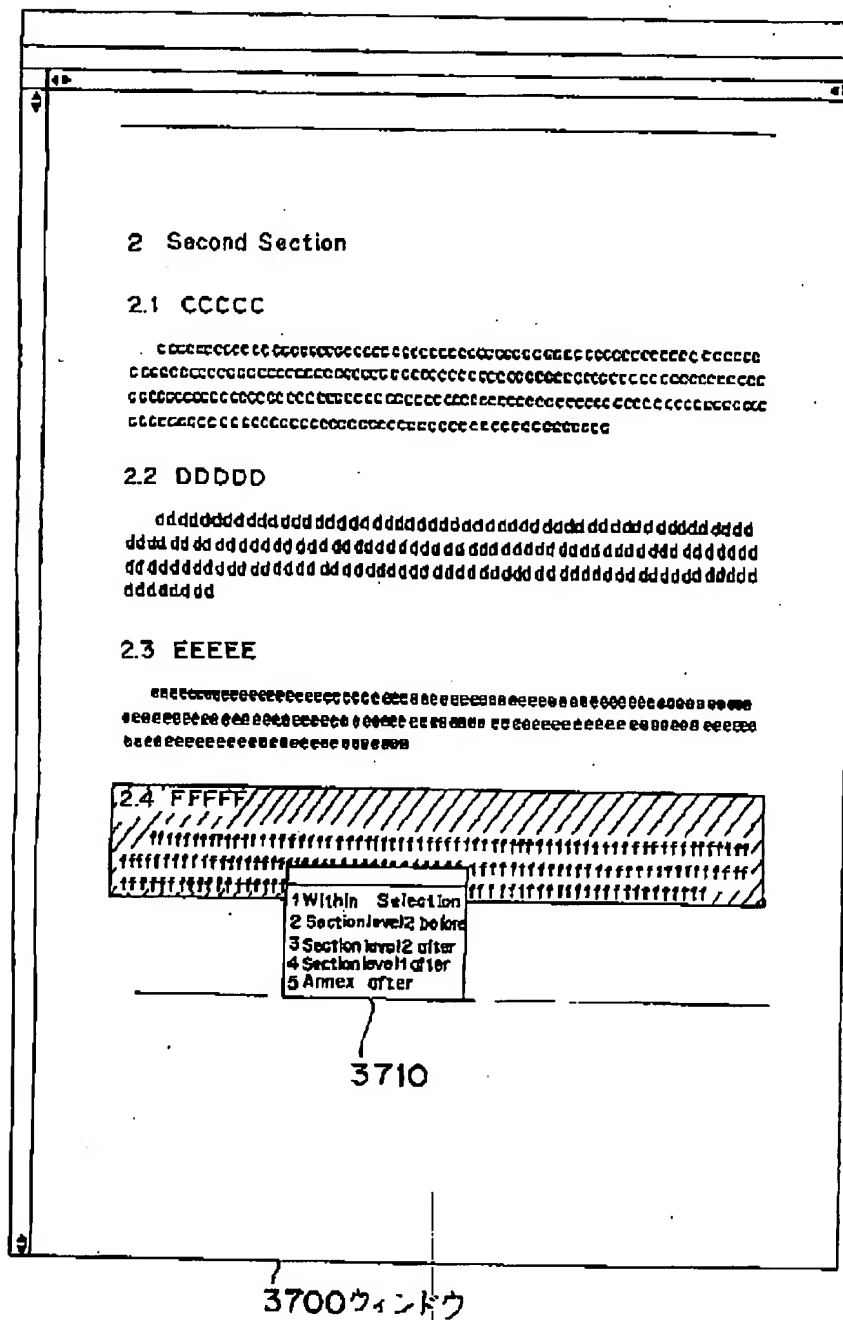
2.4 節か5は $[2.4-, 2.4.1, 2.4+, 2+]$ が追加可能

2.3 節からは $[2.3-, 2.3.1, 2.3+]$ が追加可能

(44)

特開平5-108631

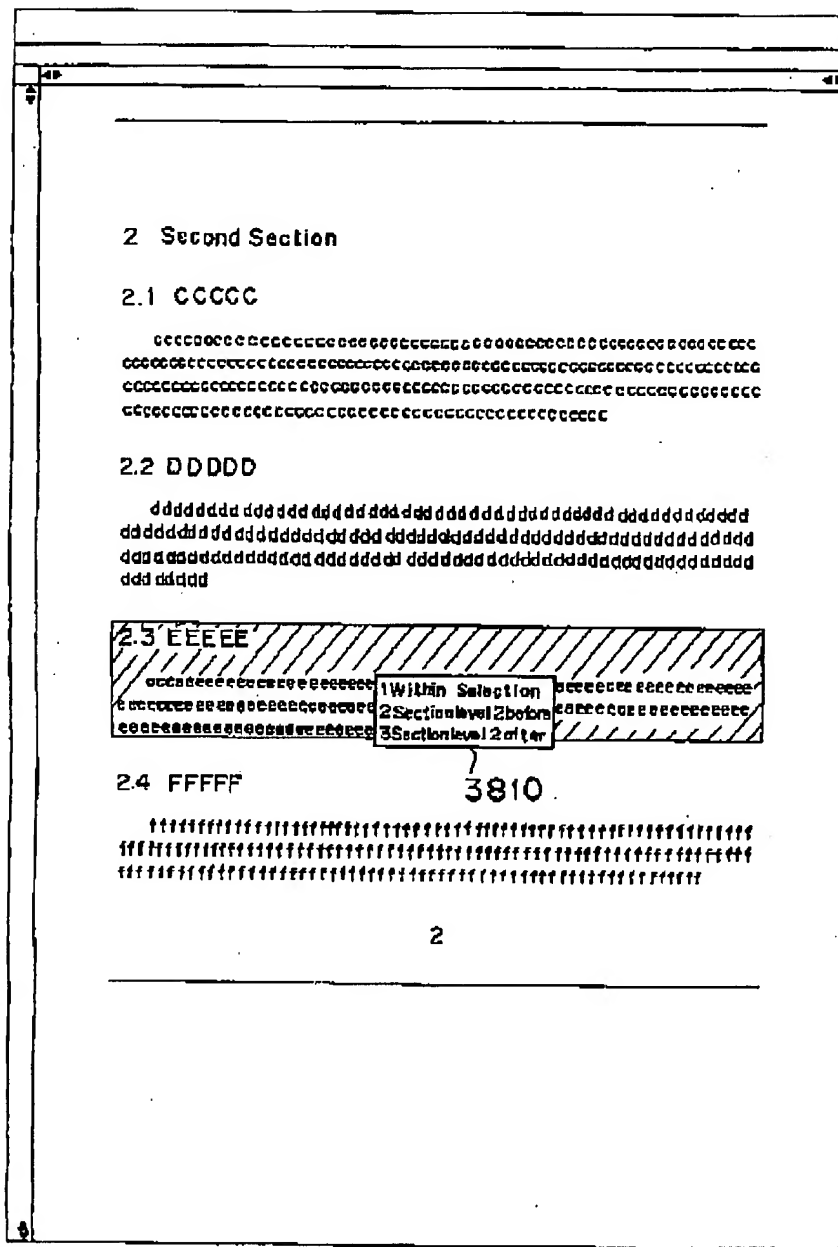
【圖 37】



(45)

特開平5-108631

【図38】



3700

(46)

特開平5-108631

【図40】

